

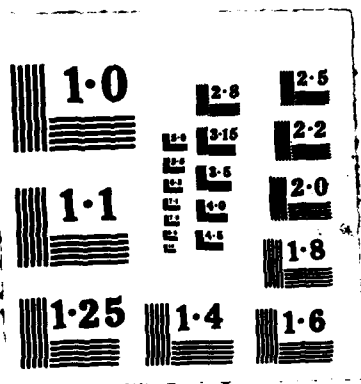
AD-A186 680

MANAGEMENT OF COMPLEX INFORMATION IN SUPPORT OF
EVOLVING AUTONOMOUS EXPERT SYSTEMS(U) AIR FORCE WRIGHT
AERONAUTICAL LABS WRIGHT-PATTERSON AFB OH.. F MARTINEZ
SEP 87 AFMAL-IR-87-1123 F/G 12/9

1/1

UNCLASSIFIED

NL

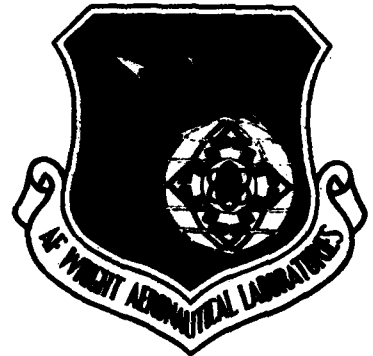


DTIC FILE COPY

AFWAL-TR-87-1123

AD-A186 680

**MANAGEMENT OF COMPLEX INFORMATION IN SUPPORT OF EVOLVING
AUTONOMOUS EXPERT SYSTEMS**



September 1987

Final Report for Period January 83 to March 87

Approved for public release; distribution unlimited.

**DTIC
ELECTE
NOV 24 1987
S H D**

**AVIONICS LABORATORY
AIR FORCE WRIGHT AERONAUTICAL LABORATORIES
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433-6543**

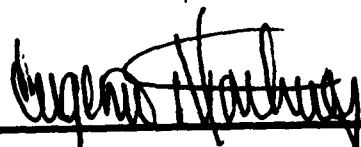
87 11 14 072

NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture use, or sell any patented invention that may in any way be related thereto.

This report has been reviewed by the Office of Public Affairs (ASD/PA) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.

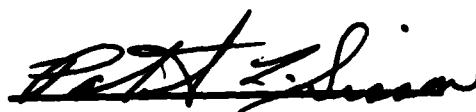


EUGENIO MARTINEZ, Capt, USAF
Project Engineer

FOR THE COMMANDER



DONNA M. MORRIS, Chief
System Evaluation Branch
Avionics Laboratory



If your address has changed, if you wish to be removed from our mailing list, or if the addressee is no longer employed by your organization please notify AFVAL/AAAF, W-PAFB, ON 45433 to help us maintain a current mailing list.

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE				Form Approved OMB No. 0704-0188										
1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED			1b. RESTRICTIVE MARKINGS N/A											
2a. SECURITY CLASSIFICATION AUTHORITY AFWAL/AAAF			3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for Public Release; Distribution is Unlimited											
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A			5. MONITORING ORGANIZATION REPORT NUMBER(S) Same as 4											
4. PERFORMING ORGANIZATION REPORT NUMBER(S) AFWAL-TR-87-1123			7a. NAME OF MONITORING ORGANIZATION Same as 6a											
6a. NAME OF PERFORMING ORGANIZATION Software Development Group		6b. OFFICE SYMBOL (If applicable) AFWAL/AAAF-2		7b. ADDRESS (City, State, and ZIP Code) Same as 6c										
8a. NAME OF FUNDING/SPONSORING ORGANIZATION Same as 6a		8b. OFFICE SYMBOL (If applicable) N/A		9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER N/A										
8c. ADDRESS (City, State, and ZIP Code) Same as 6a		10. SOURCE OF FUNDING NUMBERS <table border="1"><thead><tr><th>PROGRAM ELEMENT NO.</th><th>PROJECT NO.</th><th>TASK NO.</th><th>WORK UNIT ACCESSION NO.</th></tr></thead><tbody><tr><td>62204F</td><td>2003</td><td>02</td><td>80</td></tr></tbody></table>				PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT ACCESSION NO.	62204F	2003	02	80	
PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT ACCESSION NO.											
62204F	2003	02	80											
11. TITLE (Include Security Classification) Management of Complex Information in Support of Evolving Autonomous Expert Systems														
12. PERSONAL AUTHOR(S) MARTINEZ, EUGENIO														
13a. TYPE OF REPORT Final Technical Report		13b. TIME COVERED FROM 83-01-06 TO 87-03-31		14. DATE OF REPORT (Year, Month, Day) September 1987										
15. PAGE COUNT 73		16. SUPPLEMENTARY NOTATION N/A												
17. COSATI CODES <table border="1"><thead><tr><th>FIELD</th><th>GROUP</th><th>SUB-GROUP</th></tr></thead><tbody><tr><td>01</td><td>03</td><td>10</td></tr><tr><td>12</td><td>09</td><td></td></tr></tbody></table>			FIELD	GROUP	SUB-GROUP	01	03	10	12	09		18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number) Artificial Intelligence; Expert Systems, Knowledge Engineering; Autonomous Systems, Planning; Decision-Making; Database Management Technology; Fuzzy Theory; Category Theory		
FIELD	GROUP	SUB-GROUP												
01	03	10												
12	09													
19. ABSTRACT (Continue on reverse if necessary and identify by block number) Automata whose performance is dependent on specific information are referred to as expert systems. These effect and/or induce situations in the mission environment purposely and as supported by their respective subdomains should be able and capable of evolving concurrently with and relative to an ever evolving mission environment. Information perceived from the latter may be complex, i.e., with multivariate, interrelated and dynamic patterns. The following addresses the problem of complex information management in support of autonomous expert systems within evolving environments. Emphasis is placed on the systems ability to infer generalisations, appraise the circumstantial states of the mission environment, and perform appropriate decision-making. (Keywords:)														
20. DISTRIBUTION/AVAILABILITY OF ABSTRACT <input checked="" type="checkbox"/> UNCLASSIFIED/UNLIMITED <input type="checkbox"/> SAME AS RPT. <input type="checkbox"/> OTIC USERS			21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED											
22a. NAME OF RESPONSIBLE INDIVIDUAL Eugenio Martinez, Capt, USAF			22b. TELEPHONE (Include Area Code) (513) 255-2446		22c. OFFICE SYMBOL AFWAL/AAAF-2									

DD Form 1472, JUN 86

Previous editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE

UNCLASSIFIED

ERRATA SHEET FOR AFWAL-TR-87-1123

Page	Item/ Should Read (or Comment)
ii	Block 19/ (misprinted; see summary in page iii)
vi	Figure 5/ "Entity-Relational ..." (also in page 9) Figure 18/ "... Four-Class ..." (also in page 26) Figure 24/ "... Grid-By-Grid ..." (also in page 34)
vii	Figures 26 & 27/ "... $t_i + 3$ " Figure 28/ "... Conceptron's ..." (also in page 39) Figure 31/ "Pseudo-Pascal ..."
8	Paragraph b., from second sentence on/ "This suggests the use of cause-effect tuples with insufficient (e.g., fuzzy, incomplete, uncertain, noisy, etc) instantiation. In other words, these tuples may be context-free or context-insensitive." Next paragraph, second sentence/ "... obtaining a faster while good-enough degree of awareness under identical conditions of expertise."
9	Figure 5b/ (label "ENTITY #2" should be located under segmented line)
23	Third paragraph, last sentence/ "... individual domains (Figure 3)."
25	Title in Figure 17/ "... FOR ENVIRONMENTAL ..."
38	Second paragraph, last sentence/ "The PLANNER's Conceptron scheme is ..." Third paragraph, last sentence/ "...sequence of desired or ideal mission ..."
39	Figure 28's right side/ (all implicating parentheses "{" should face outwards "}"; bottom three are not; two parentheses in the middle should each span respective "Deviation" and "Traversability" class pair)
40	Last sentence/ "... pseudo-Pascal ..."
43	Title in Figure 31/ "PSEUDO-PASCAL VERSION OF PILOT'S CONCEPTRON"

(cont'd)

AD-A186680

ERRATA SHEET FOR AFWAL-TR-87-1123

Page	Item/	Should Read (or Comment)
62	First sentence/	"... of this vehicle's theoretical ..."
63	Reference #2/	"... Conceptual Design ..."

SUMMARY

Automata whose performance is dependent on specific information are referred to as expert systems. These effect and/or induce situations in the mission environment purposely and as supported by their respective subdomains of awareness, decision and response. Therefore, these expertise subdomains should be able and capable of evolving concurrently with and relative to an ever evolving mission environment. Information perceived from the latter may be complex, i.e., with multivariate, interrelated and dynamic patterns. The following addresses the problem of complex information management in support of autonomous expert systems within evolving environments. Emphasis is placed on the systems ability to infer generalizations, appraise the circumstantial states of the mission environment, and perform appropriate decision-making.



Accession For	
NTIS GRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	

FOREWORD

This document describes work performed by the author at the Autonomous Machine Intelligence Laboratory, Department of Electrical Engineering, University of Florida, Gainesville, for the United States Army and the Defense Advanced Research Project Agency, during the period 6 January 1983 to 27 November 1984.

The work was revised during the period 20 January 1985 to 31 March 1987, at the Cybernetics Section of the Software Development Group, System Avionics Division, Air Force Wright Aeronautical Laboratories, Wright-Patterson Air Force Base, Ohio, under in-house project 20030280, Cognitive Processing Techniques.

This report constitutes an initial and modest portion of a more comprehensive effort to explore and apply artificial evolutionary techniques in the quest for the realization of an unnatural living state. Other related reports will follow in the future.

TABLE OF CONTENTS

SECTION	PAGE
1.0 INTRODUCTION	1
1.1 The Concept of Expertise Revisited	1
1.2 Enhancing the Expertise	3
2.0 THE AUTONOMOUS EXPERT VEHICLE AND ITS DISTRIBUTED SPECTRUM OF INFORMATION	8
2.1 Mission Awareness	8
2.2 Managing Awareness	18
3.0 REQUIREMENTS IN SUPPORT OF DECISION-MAKING AND RESPONSE	28
3.1 The Generalized Conceptor	28
3.2 Applied Decision-Making and Response	31
4.0 AWARENESS THROUGH THE MISSION MAPS	44
4.1 Logical Design and Suggested Implementation	44
4.2 Access to the Information in the Mission Maps	49
5.0 SIMULATION RESULTS	52
6.0 CONCLUSION	62
REFERENCES	63
BIBLIOGRAPHY	64

LIST OF ILLUSTRATIONS

FIGURE		PAGE
1.	Conventional Expert System Architecture	2
2.	Autonomous Expert System Architecture	4
3.	Autonomous Land Vehicle Multi-Domain Concept	5
4.	Logical Spectrum Of Information For Single Domain	7
5.	Entity Relational Models	9
6.	Snapshot Of Multivariate Awareness State At Time t	10
7.	Generalized E-R-E Model For Evolving Mission/Environment	12
8.	Awareness On Mission Entities Along Time	13
9.	Snapshot Evolution vs. Planned Mission	14
10.	Domain Of Expertise's Focus Of Attention (Decision-Making)	15
11.	PLANNER's Mission Map	16
12.	NAVIGATOR's Mission Map	17
13.	Interdomain Communication Protocol	19
14.	MANAGER's Dictionaries	20
15.	Intradomain Use Of Dictionaries	21
16.	Library Template Composition	22
17.	Logical Template For Environmental Entities	25
18.	Sample Four Class Template	26
19.	Instantiation Samples	27
20.	Conceptron Concept	29
21.	Sample Inferential Chain	30
22.	Sample Mission Scenario: Cartesian Grid Overlay	32
23.	Nondynamic Traversability Map (Binary)	33
24.	Preliminary Grid By Grid Traversability	34
25.	Preliminary Forecasting Of Dynamic Entities	35

LIST OF ILLUSTRATIONS (CONTINUED)

FIGURE		PAGE
26.	Dynamic Map Forecasted For T_i+3	36
27.	Generalized Traversability Map For T_i+3	37
28.	Initial Conceptron Association Weights For NAVIGATOR	39
29.	PILOT's Vocabulary	41
30.	Matrix Representation Of PILOT's Conceptron	42
31.	Pseudo PASCAL Version Of PILOT's Conceptron	43
32.	Sample Mission Scenario: Original Map	46
33.	Sample Mission Scenario: Evolution Through Octal-Grid Displacement	47
34.	Network Scheme	48
35.	Sample Relational Tables	50
36.	Simulation Scenario	53
37.	Initial Move	55
38.	Binary Traversability for First Goal	56
39.	Second Goal	58
40.	Third Goal	59
41.	Final Goal	60
42.	Final Results	61

1.0 INTRODUCTION

"...battle is not a football game with players in fixed positions; battle should be like soccer with rapidly shifting play across the field, and autonomous decision-making by the players..."

AirLand Battle 2000

1.1 The Concept of Expertise Revisited

Artificial Intelligence in general centers around the mimicking of Nature's traits, in specific, observable information processing. Mainstream (or conventional) Artificial Intelligence is currently aimed at the exploration and development of the popularly termed expert (or production) systems. These may be considered purposeful systems targeted for a predictive and effective performance based on a limited but well established domain of awareness. Their primary role is one of decision support.

Unfortunately, the concept of expert systems is usually defined after the popular techniques with which it is forged. Most implementations follow the predicate calculus approach: discernment and elicitation are formulated through "if-then-else" rule sequences. Perhaps because of this, the generic architecture (1) suggests three fundamental and physically distinct functional domains: data (operands), control (rules), and operations (actions) (Figure 1). The data domain supplies the operands to the rule chain embedded in the control domain. The latter then commands the operation domain to execute some response upon operand admissibility. The separation of these functions is evident in conventional expert system shells (developmental tools) and executable production-quality packages.

The design to be suggested in this paper differs not from the mainstream understanding of expert system performance but rather from that of implementation. For example, expert systems must be categorized as automata managing specific information in order to perform a particular mission, "information management" labeled independently of the means. The designer must approach applications with the philosophy of satisfying mission requirements, not market product specifications. That is, the design must conform to the requirements of the application, not to the specifications of the product with which the application will be implemented.

The expert system to be presented in this paper is logically designed with the above philosophy in mind. It is basically an architecture in which the otherwise loosely coupled mainstream design synthesizes the expertise domains (operands, rules and actions) into an object-driven processing "monolith". This novel approach, classifiable under the realm of linear-decision functions, is called Conceptron (2,3,4,5).

The basic Conceptron model integrates "weighted" associations between cause-effect tuples. The latter make up the input and output entries, respectively, to and from the object-driven network of linear functor combinations, where the weighted associations are realized. Based on modified entity-relational (6,7) and causality models (40), the Conceptron approach is better suited to handle the complex information burden

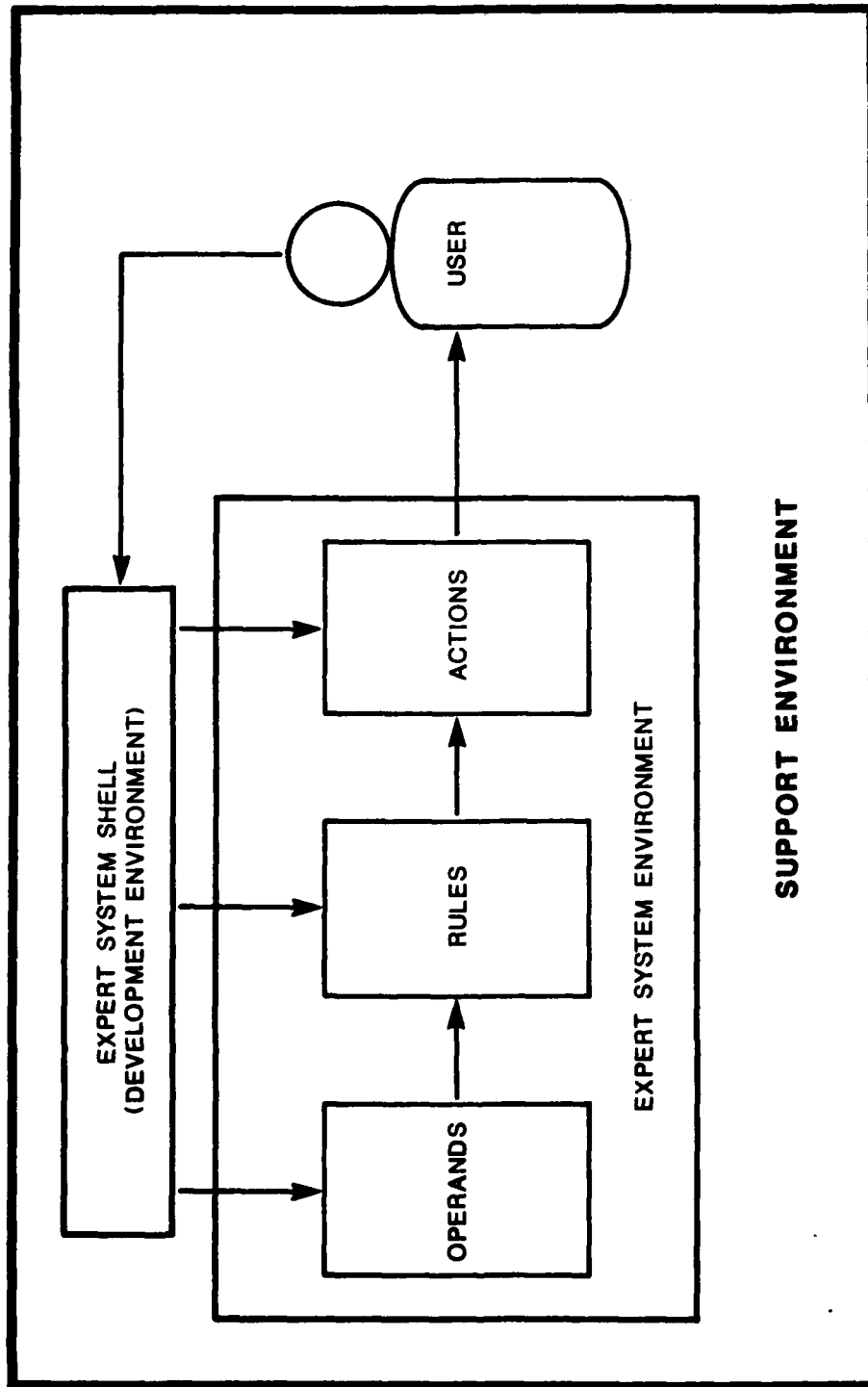


Figure 1. CONVENTIONAL EXPERT SYSTEM ARCHITECTURE

expected in the execution of high-throughput autonomous decision-making and response. Hereinafter, the terms object and entity are used interchangeably unless noted otherwise.

Invariably, operands, rules, and actions characterize the expertise embedded in a given mainstream expert system. Similarly, the object space classified and generated by the Conceptron spans the autonomous expert system's expertise. In both cases, the resulting performance will be dependent on the quality of that expertise, regardless of its designation (i.e., whether operands, rules, actions, or objects). Ideally, to an observer, the resulting performances should be indistinguishable if the two different expert systems are furnished with similar expertises.

However, this is only true if the expert systems are exercised in their respective ideal testbeds. Once a given design operates in a less-than-ideal environment, its generic traits are brought to the limelight. Common sense would tend to favor that approach which appropriately conforms to the logical structure of the situation it operates within; that approach which implies a lesser deviation from reality, if reality is a transient one. Dynamic object-oriented architectures are better equipped than mainstream ones to support expert systems confronted with this type of requirement. Therefore, an improved performance over conventional architectures is expected from a Conceptron-like expert system.

1.2 Enhancing the Expertise

Expert systems technology is suitable for supporting applications where the utility of human resources is not justifiable, i.e., cost-ineffective. Usually these applications are of aerospace or military type with either very high or very low manned dependence, with too-stressful-to-human-safety considerations. In these extreme cases, the conversion of user-assisted expert systems into autonomous (i.e., machine-only) ones becomes not only the unique solution but also the challenge. The mainstream expert system architecture, as is, cannot efficiently (and even effectively) support the requirements found in these types of applications unless a self-supporting capability is provided.

The recurring limitations in mainstream expert systems are self-imposed. The expert systems approach in Conceptron counters this by integrating a perception subdomain (Figure 2). The integrated product is henceforth termed an autonomous expert system. Real perception of the environment, being a function delegated to and collectively performed by multiple sensors, is of utmost interest; however, it is outside the direct scope of the design to be presented here. The problem of internal data management, with respect to that of compiling (or fusing) the sensorially provided information and subsequently updating the expert system's overall expertise, is the subject of concern here.

The autonomous expert system under consideration in this paper is a land vehicle exhibiting strategic planning, tactical navigation and guidance, and closed loop piloting, each as separately distributed but interdependent domains of expertise (Figure 3). While executing a given mission, the vehicle must assess situations en route by means of preloaded ideal entity

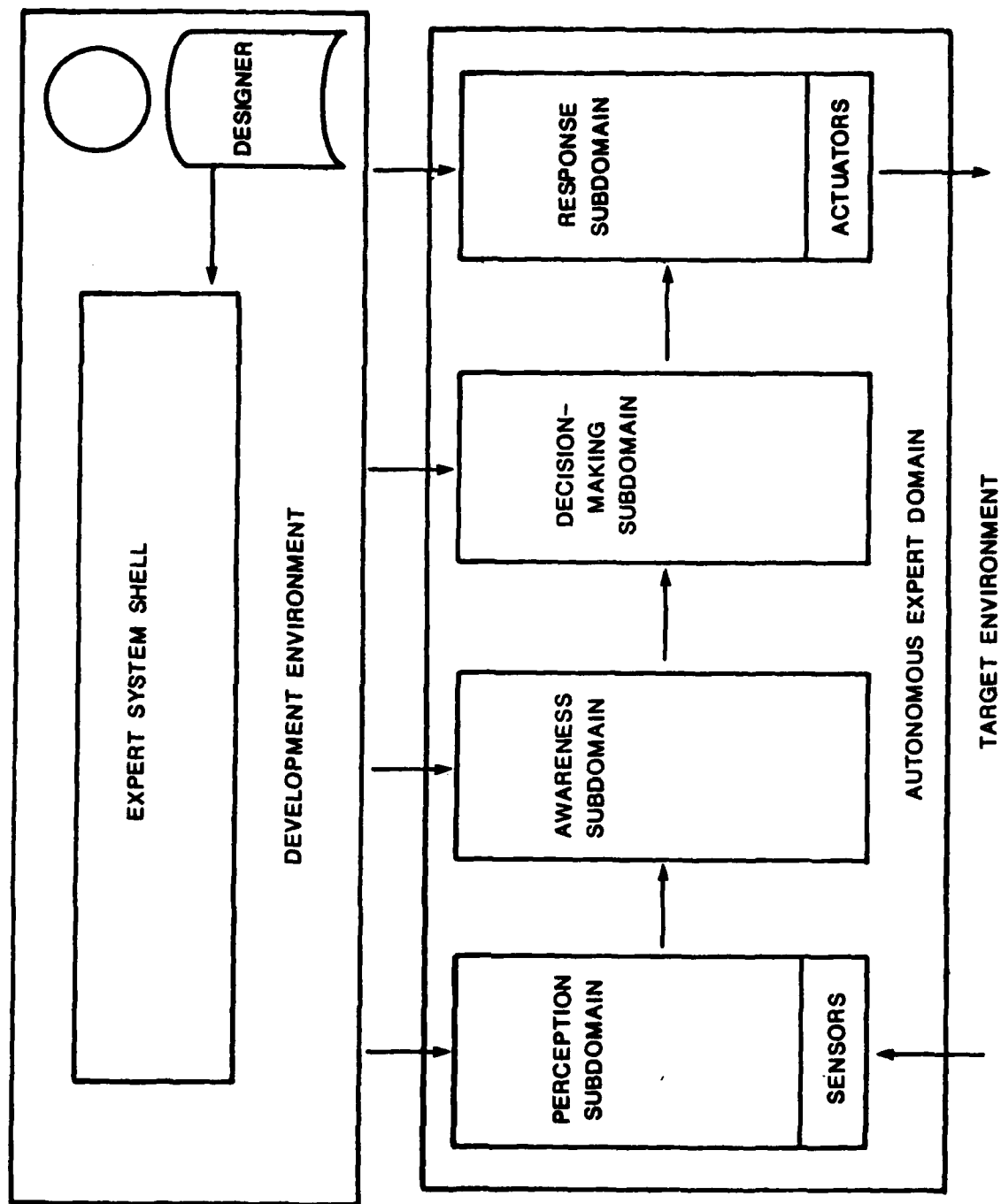


Figure 2. AUTONOMOUS EXPERT SYSTEM ARCHITECTURE

templates and real-time perception-based instance updating. Then, based on the embedded tasks and expected performance indexes, it must decide on and follow an "appropriate" mission evolution. This includes developing contingencies for unexpected hazards and alternate goals. Decision-making is based on a combination of cost-functions incorporating, e.g., minimum distance and minimum time deviations.

The first domain of expertise, the PLANNER, reschedules mission milestones as necessary. The second domain of expertise, the NAVIGATOR, finds a near-optimal path (sequence of passageways) connecting contiguous mission milestones. The third domain of expertise, the PILOT, determines actual locomotion and steering in order to follow the path. Each of these processes utilizes its own version of a generalized mission map continuously updated by variable range and resolution sensors. This trio or assemblage of maps is centrally administrated by a fourth domain, the awareness MANAGER.

The scope of this paper focuses on the logical design of an awareness function conformable to the mission environment without sacrificing its support to the vehicle's mission (Figure 4). This design will be based on a modified Conceptron architecture.

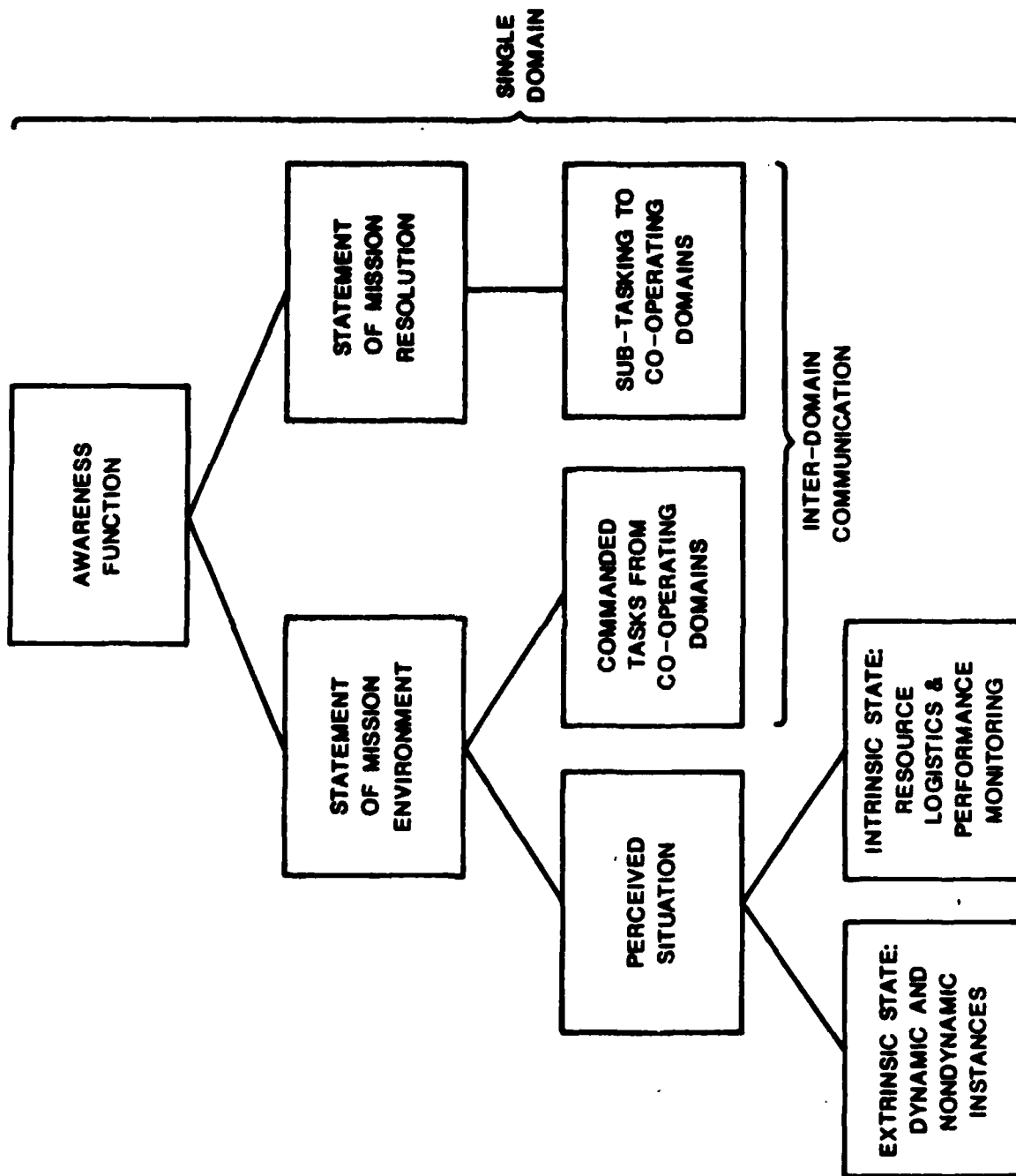


Figure 4. LOGICAL SPECTRUM OF INFORMATION FOR SINGLE DOMAIN

2.0 THE AUTONOMOUS EXPERT VEHICLE AND ITS DISTRIBUTED SPECTRUM OF INFORMATION

2.1 Mission Awareness

Taking into consideration the physical nature of the information to be managed (both linguistical* and pictorial types) the following discussion will be from a rather logical perspective. Using a modified Entity-Relational (E-R) model (Figure 5), situations within the vehicle's mission (10) envelope will become classifiable under E-R sets (lattices) and sets of E-R sets, regardless of whether symbolic or pictorial origin. The Entity-Relational-Event (E-R-E) model is introduced to distinguish dynamic from nondynamic relations among entities. This will be explained later in the paper. Hereinafter, the terms set and lattice will be used interchangeably unless noted otherwise.

One of the primary objectives in this work is that of enhancing the reliability of the vehicle's overall performance by complementing its fundamental (preloaded) knowledge through a high-level awareness of both perceptible and intangible circumstantial evidence (Figure 6). The balance between how much and what (quantity and quality of) information to manage becomes crucial since the vehicle's performance throughput depends on it. Throughput efficiency starts with context representation. There are two likely approaches to context representation:

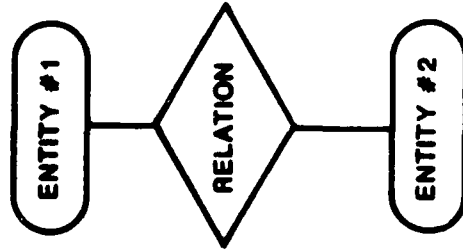
a. the overall mission map could be loaded with specific information about "everything" that the vehicle may ever come across. This way the vehicle would respond in accordance to the predetermined cause-effect tuples whenever coming across a real match within the mission environment; or,

b. the vehicle could be allowed extreme flexibility and would succeed better in a complex environment if relaxed to do "appropriate" decisions (measured against predetermined criteria) instead of searching its memory for an exact precedent to control its responses. This does not preclude the use of cause-effect tuples which may be insufficiently instantiated. In other words, these tuples cannot be context sensitive.

Given that the mission scenario is unknown and random (entities may be recognizable against probabilities of certainty and expectancy) at the mission's onset, the "appropriateness" approach becomes the main doctrine behind the design proposed in this paper. The reason being its improvement in information entropy over the exhaustive method, i.e., obtaining a higher degree of awareness. To the degree a given domain of expertise maximizes useful information yield from a minimum of just instanced information, both its awareness and, unfortunately, its structural complexity increases. The expert vehicle's understanding is prejudiced by the quantitative and qualitative incompleteness inherent in a given domain of expertise.

* note: it is assumed that the PILOT domain could be replaced by a human pilot, in which case verbal communication may be considered. Also, the vehicle may communicate verbally with otherwise manned vehicles and systems within or outside the mission environment.

a. BASIC E-R MODEL



ALL INSTANCES

b. MODIFIED E-R MODEL (E-R-E)

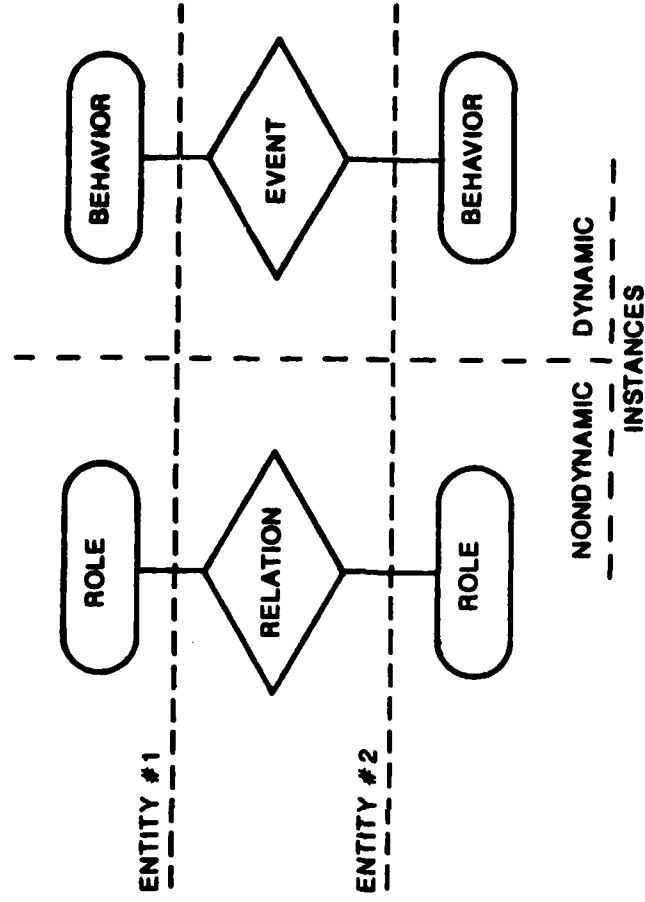


Figure 5. ENTITY RELATIONAL MODELS

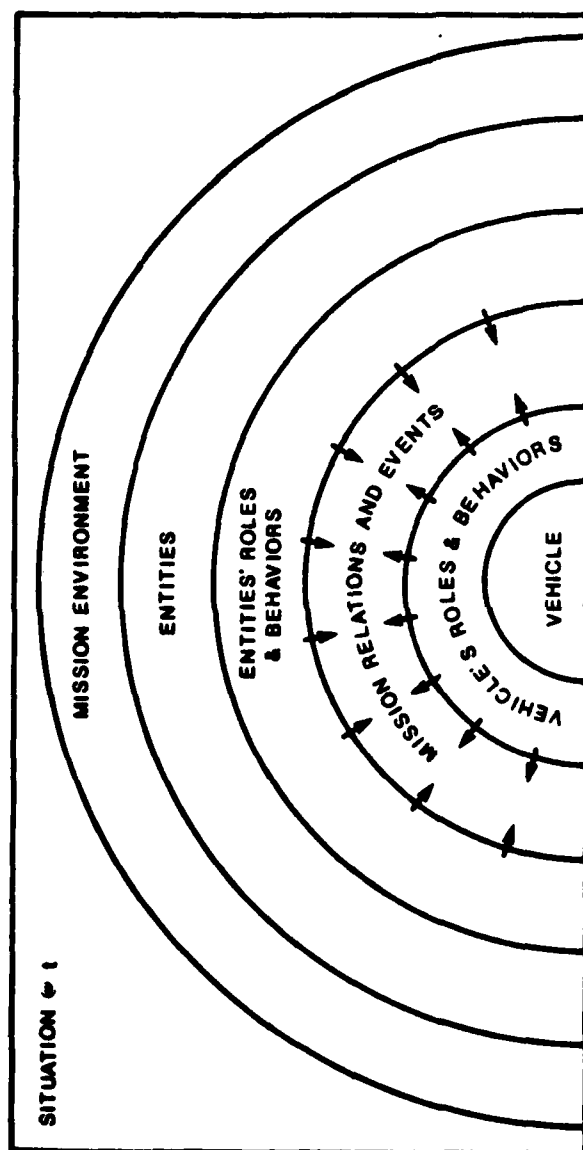


Figure 6. SNAPSHOT OF MULTIVARIATE AWARENESS STATE AT TIME t

The logical synthesis of the information contained in the vehicle's mission map assemblage supports the concerted awareness that the distributed expert domains must have about the mission envelope and its surrounding environment. Therefore, the map assemblage must, as loyally as possible, capture real situations beyond a limited spatiotemporal dimensionality (4D) by classifying entities (objects and events) and their circumstances (traits and interactions) as additional temporally evolving variables (Figures 7:9). Therefore, the mission envelope and its environment may be parametrically and internally regenerated by the vehicle as a linear hypersurface representing an augmented time-dependent E-R lattice.

As mentioned, this hypersurface constitutes the vehicle's multivariate conception of the complex environment that forms and surrounds its mission envelope. The vehicle is considered another entity within this complex reality, and for this reason its instantaneous Focus of Attention (FoA) is on a localized area within that reality (Figures 10:12). The FoA varies with each individual domain of expertise within the vehicle's assemblage. The PLANNER has a cumulative global or strategic view of the perceived mission scenario. The NAVIGATOR and the PILOT have instead a regional (tactical) and a local (immediate) view, respectively, of the situation surrounding the vehicle during a given time period and at a given instant, respectively.

In other words, and from the different FoAs, the overall vehicle awareness "transits" throughout the mission scenario as the circumstances in the mission envelope evolve. This evolution may be treated internally by the vehicle as a sequence of mission states with its corresponding tally of accomplished and still-to-go milestones and other status information. This transition may also be interpreted by a dynamic subhypersurface evolving within the previously mentioned hypersurface. Since the hypersurface is parametrically generated, it is traceable. Therefore, the domains of expertise are able to follow time-dependent events and relations through class and instance variables.

Another important aspect is that the vehicle's self-awareness (about its intrinsic resource and performance status parameters) must be relatively comparable to its overall awareness about the mission environment. Self-awareness ranks at the highest priority among all other vehicle functions as a preventive measure in support of mission survivability. Overall vehicle performance is critically dependent on this fundamental distinction.

The vehicle's awareness about both the mission environment and itself is facilitated by its disassembly onto the three arbitrarily chosen domains of expertise: PLANNER, NAVIGATOR, and PILOT. That is, disassembly could have resulted into more or less domains of expertise if the design criteria demanded so. In any case, disassembly of a given domain into more domains is limited by the appropriateness of the information resolution at the levels of maximal simplification within the resulting mission map assemblage.

Each domain's mission map represents a subenvelope of the overall mission envelope. Each contains a dynamic spatiotemporal volume over which the particular domain of expertise can exert control. That is, the domain "knows" or is aware of whatever occurs and/or evolves (e.g., threats,

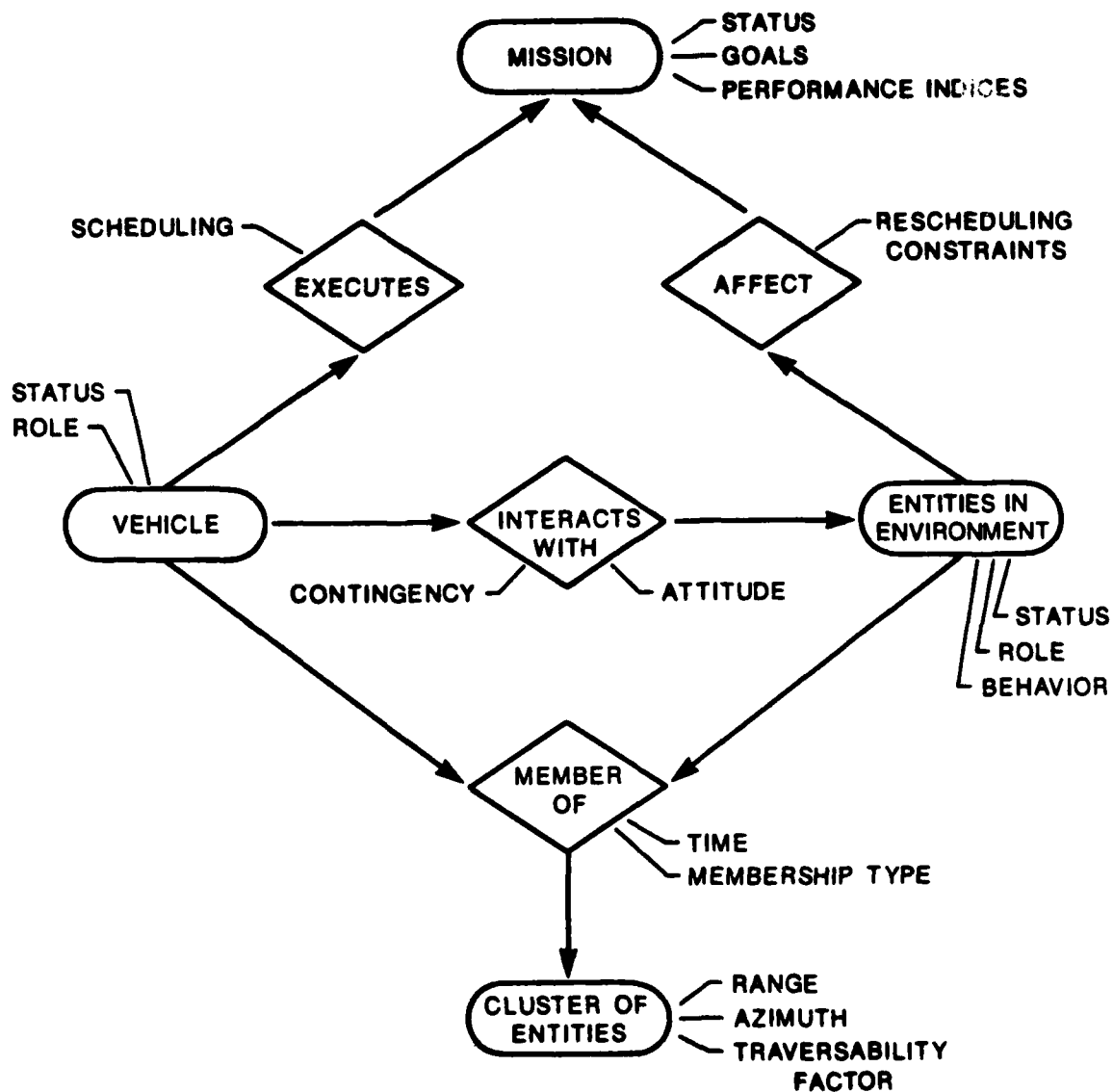


Figure 7. GENERALIZED E-R-E MODEL FOR EVOLVING MISSION/ENVIRONMENT

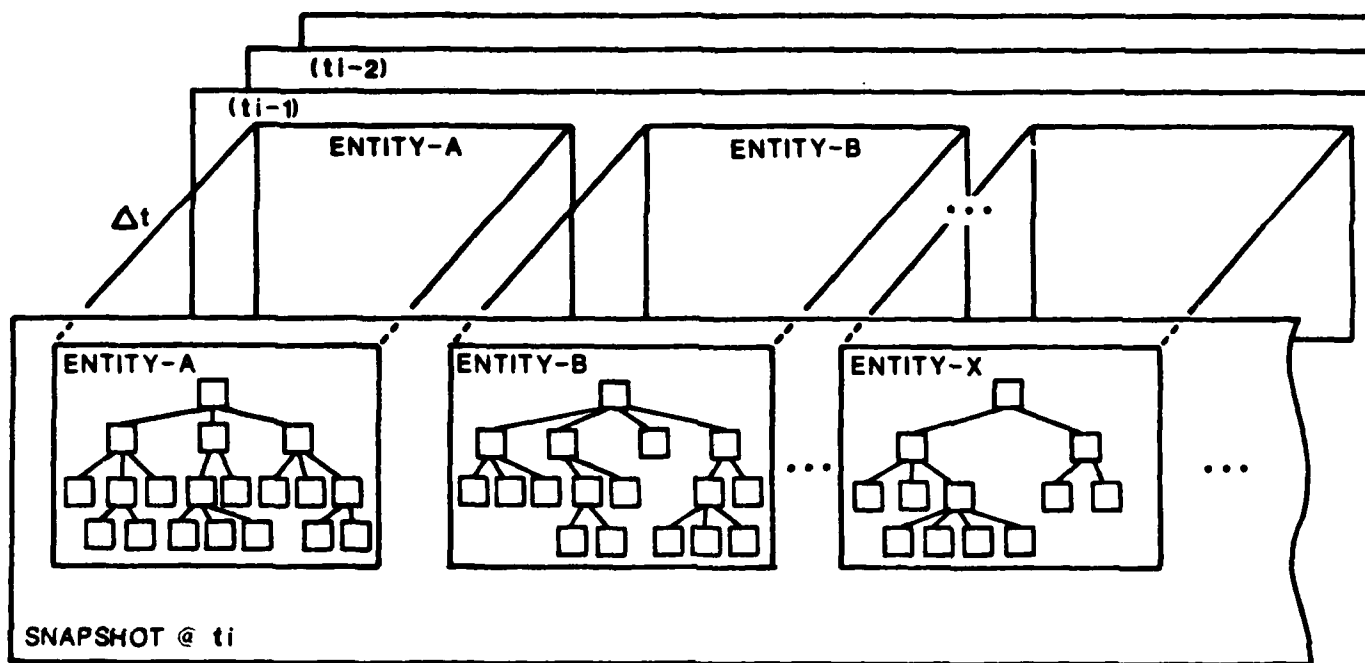


Figure 8. AWARENESS ON MISSION ENTITIES ALONG TIME

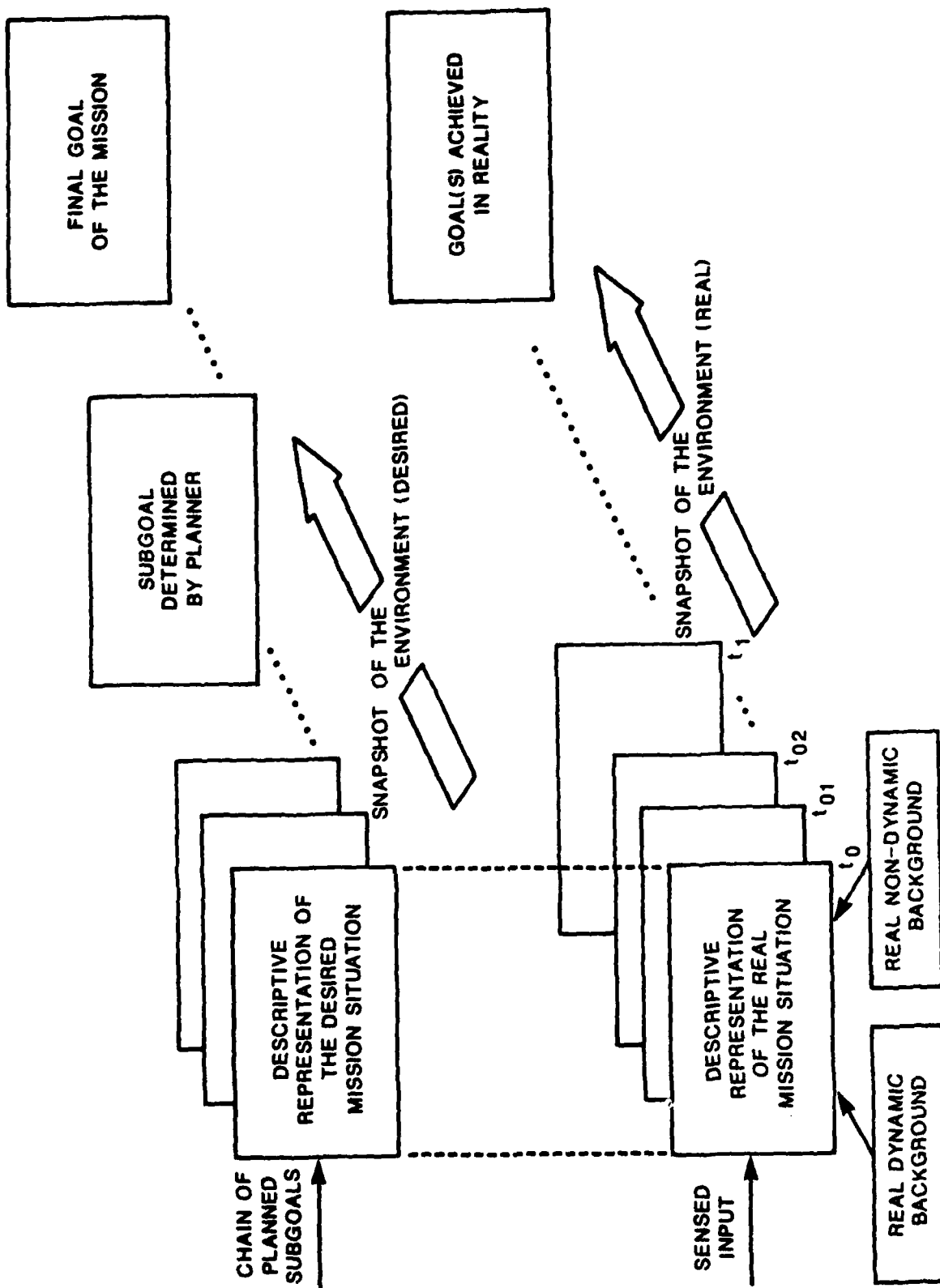
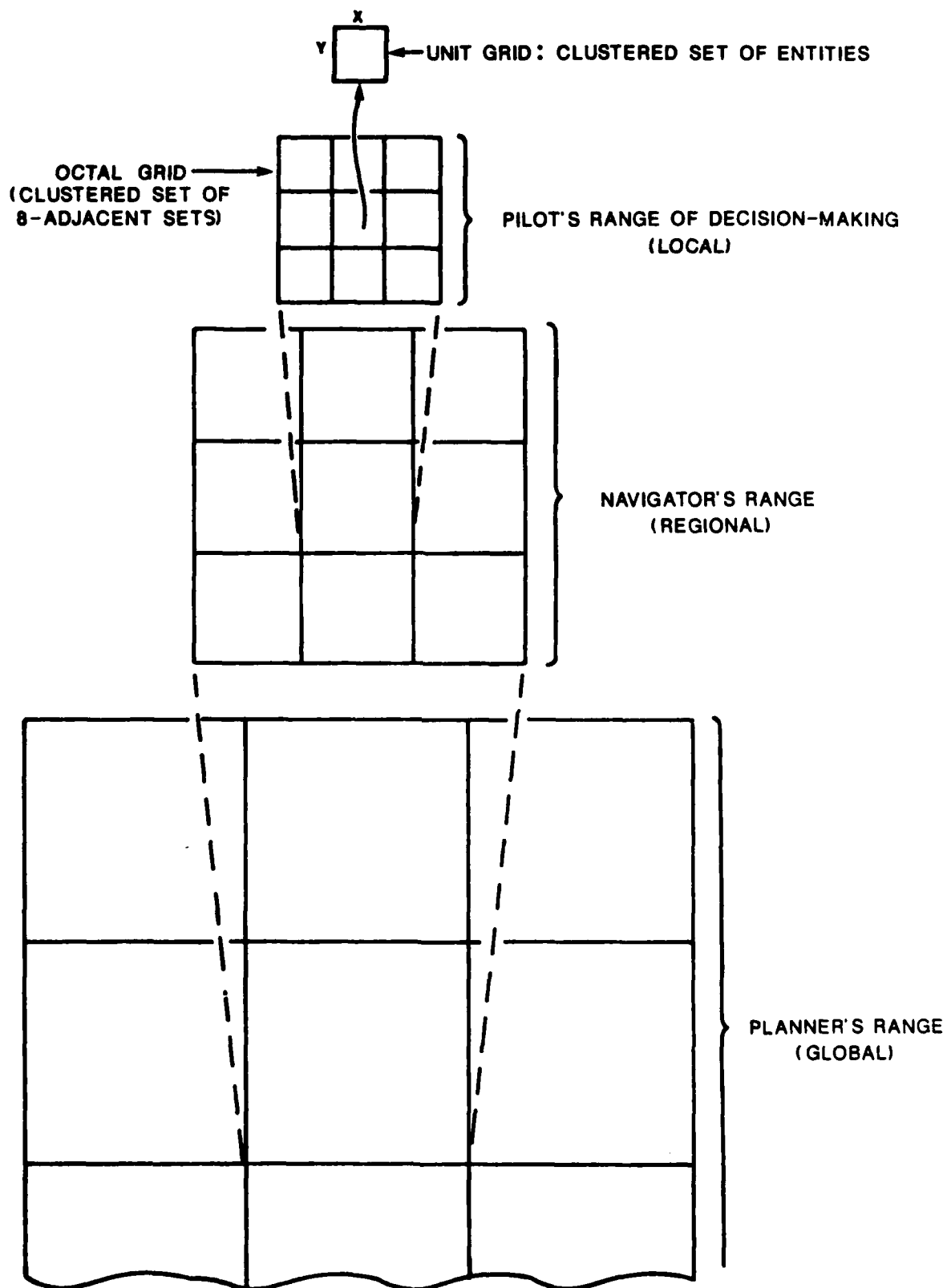
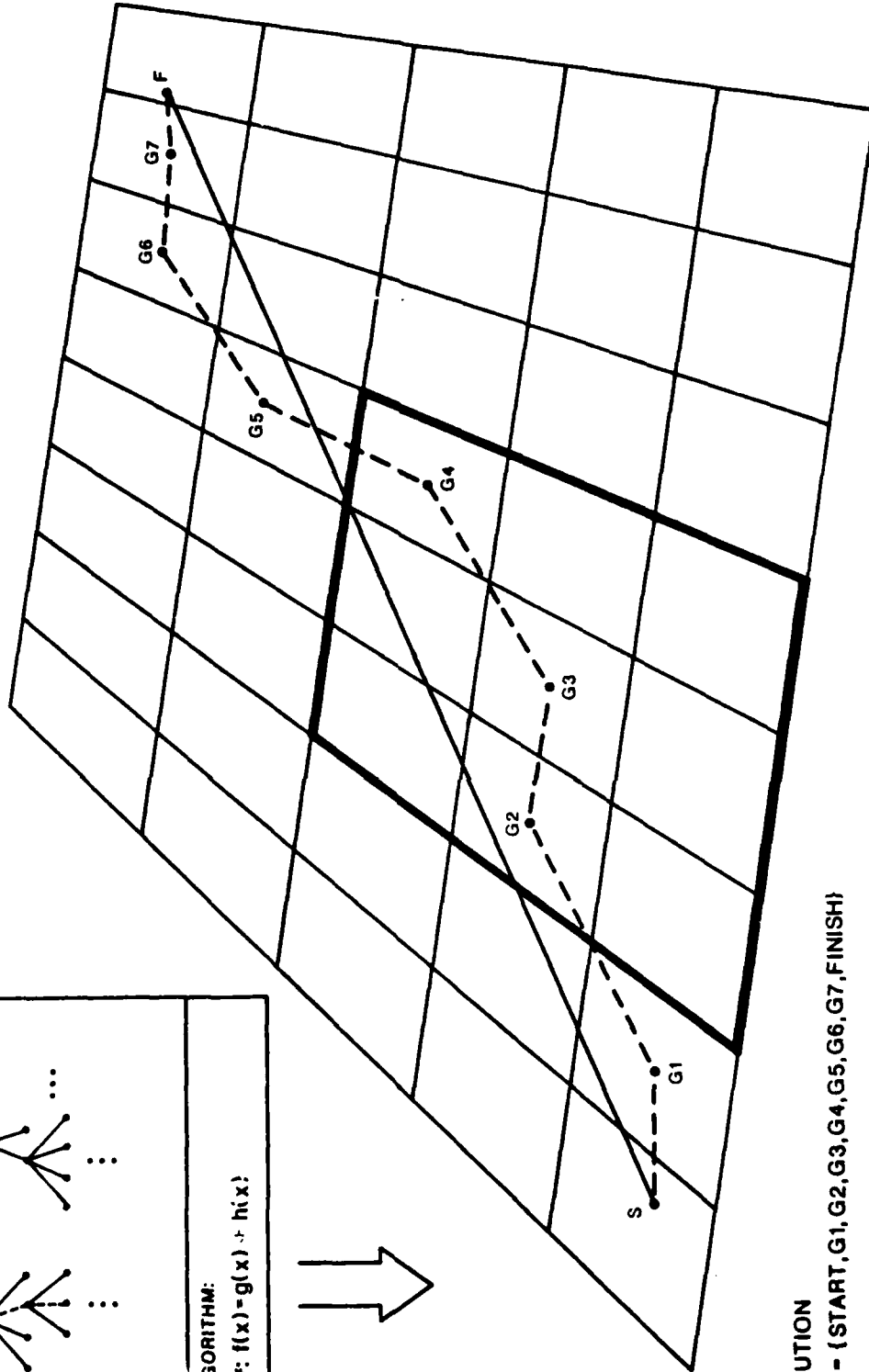
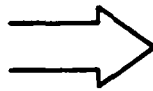
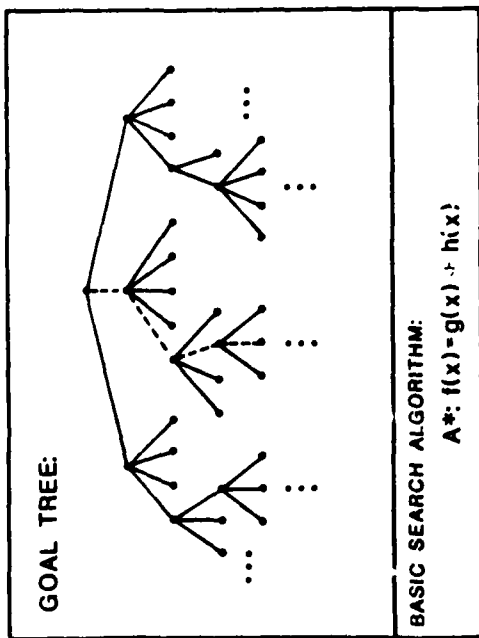


Figure 9. Snapshot Evolution vs Planned Mission



**Figure 10. DOMAIN OF EXPERTISE'S FOCUS OF ATTENTION
(DECISION-MAKING)**



PROBLEM → SOLUTION
 (START, FINISH) - {START, G1, G2, G3, G4, G5, G6, G7, FINISH}

Figure 11. PLANNER'S MISSION MAP

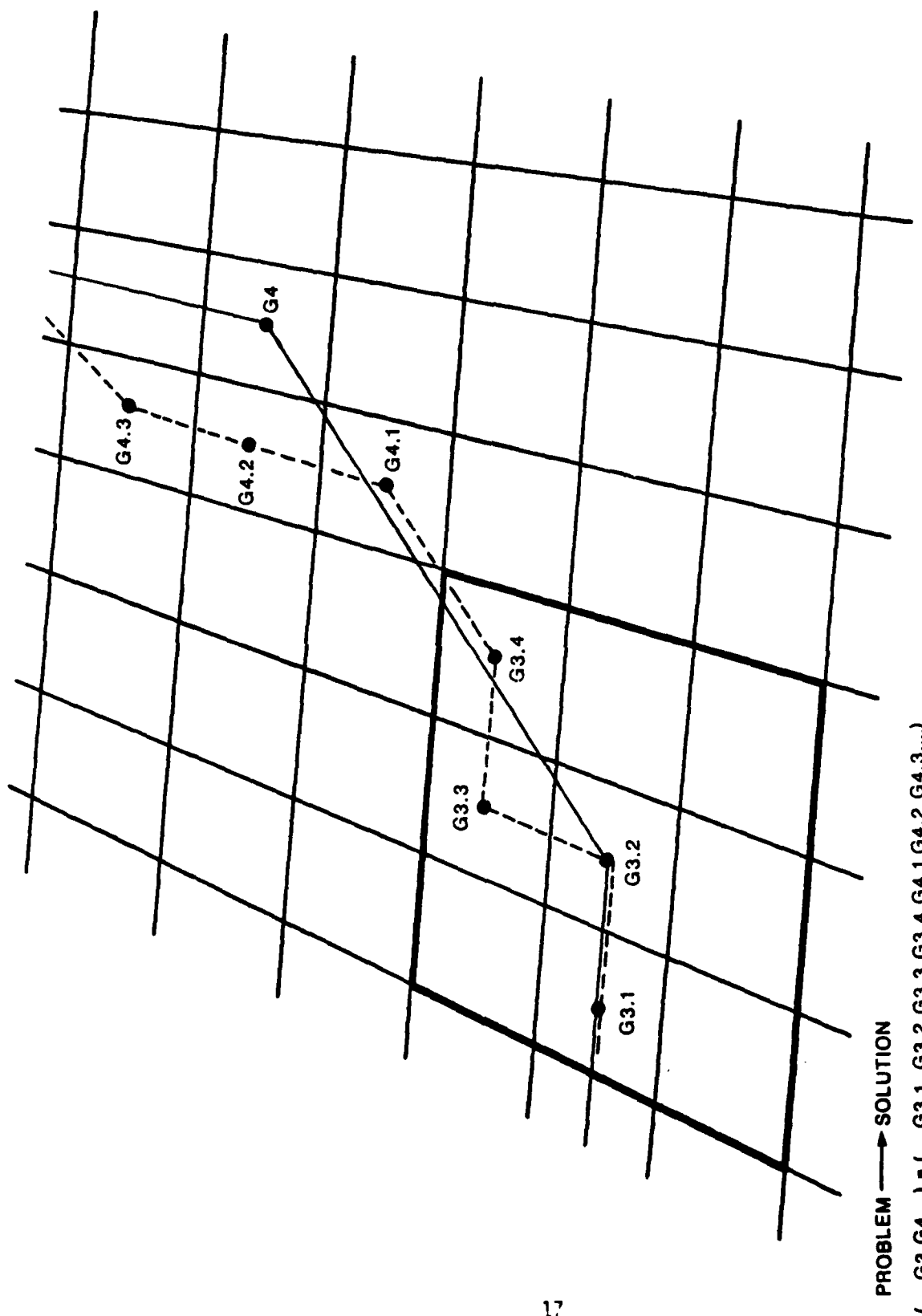


Figure 12. NAVIGATOR'S MISSION MAP

milestones) within its range of perception in time to react to it (e.g., through offensive or defensive roles).

The mission map trilogy must then represent information with granularity and coherence to the satisfaction of the awareness requirements at each individual domain of expertise without stepping beyond or short of context. In other words, the decisions and responses generated by each domain are supported by a particular and local mission map. A given domain is able to decide and act on the problem being tasked by another domain as well as following its own mission plan with the help of the MANAGER.

2.2 Managing Awareness

The vehicle's awareness architecture must conform to the natural structure of the environment as much as possible. It must also be logically decomposed onto the assemblage of three localized maps, each being particular to its respective domain of expertise. Database management technology (DBMT) (8) proves handy when trying to describe this type of distributed information problem, specially if a centralized approach to information management is opted for. There is a difference between centralized management and distributed storage. The DBMT terminology used to define such an assemblage of maps refers to distributed access through "external views."

Each access is based on a particular set of requirements, i.e., a partial view of the comprehensible environment. Again, the PLANNER, NAVIGATOR, and PILOT domains are aware of the same space and time references, but each interprets the circumstances differently in accordance with their individualized mission subenvelopes. These domains differ in context in the degree that a domain with a shorter range of awareness constitutes a subdomain within the next larger domain. Therefore, e.g., for a unique spatiotemporal reference, the PLANNER is aware of a more comprehensive envelope describing the state of the mission than the NAVIGATOR is, but the PLANNER's awareness is not as accurate as the NAVIGATOR's for that subenvelope of the overall mission envelope which is acknowledged by both. What a shorter range domain sacrifices in awareness quantity it gains in awareness quality.

Context integrity must therefore be maintained and guaranteed if the map assemblage is to perform as a whole. A background domain is to be added to the vehicle's architecture in order to support effective awareness of the mission state space at hand. In this case, the MANAGER oversees overall information acquisition and distribution with respect to the map assemblage, including intermap updating and exchange among the domains of expertise (Figures 13:16). In DBMT terms, the MANAGER acts as a database administrator.

Information individually perceived by each domain is concurrently and collectively arbitrated upon by the MANAGER. That is, the MANAGER keeps track of the vehicle's awareness by means of centralized protocol and mapping to and from each of the three maps. Real situational instances could be perceived differently by each of the separate perceptual subdomains. In that case, although collectively perceived in three ways, each instance is unique to the MANAGER.

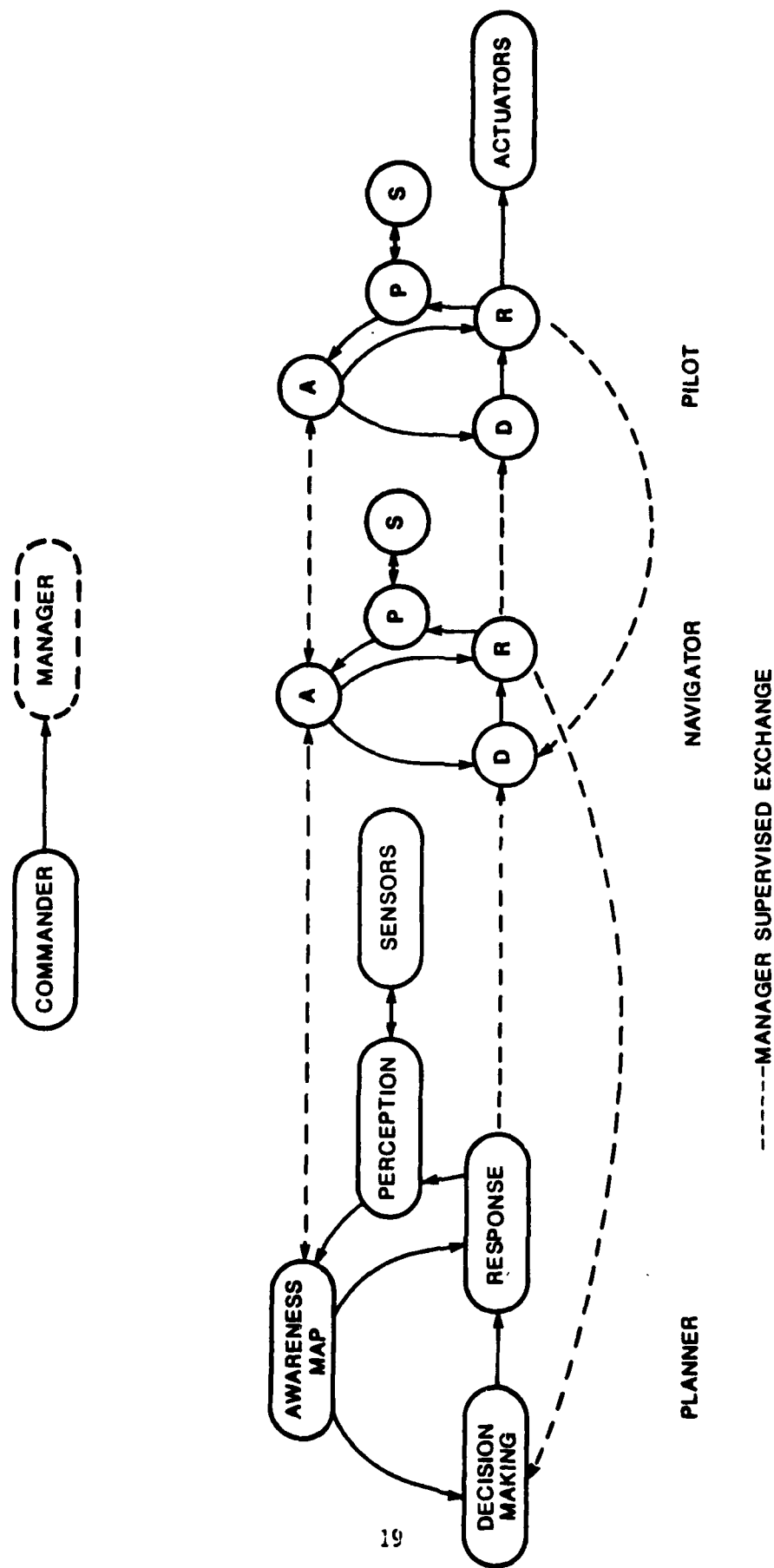


Figure 13. INTERDOMAIN COMMUNICATION PROTOCOL

DICTIONARY CLASSES	DICTIONARY ITEMS	DICTIONARY TYPE	ASSOCIATION SETS (BINARY)
①	ATTRIBUTE ENTRIES	RELATIONAL	ENTITY → ATTRIBUTES
②	INTRA-ENTITY	HIERACHICAL	ENTITY → SUBENTITIES
③	INTER-ENTITY	NETWORK	ENTITY → ENTITY → ATTRIBUTES
④	ENTITY EVOLUTION	TEMPORAL	ENTITY (t+1) → ENTITY (t)
⑤	PROBLEM SYNTHESIS	HEURISTIC	ASSUMPTION → ENTITY → ATTRIBUTES
⑥	CONCEPTRON (PROBLEM-SOLUTION)	MISSION	PROBLEM CLASS → CONTINGENCY ACTION

Figure 14. MANAGER'S DICTIONARIES

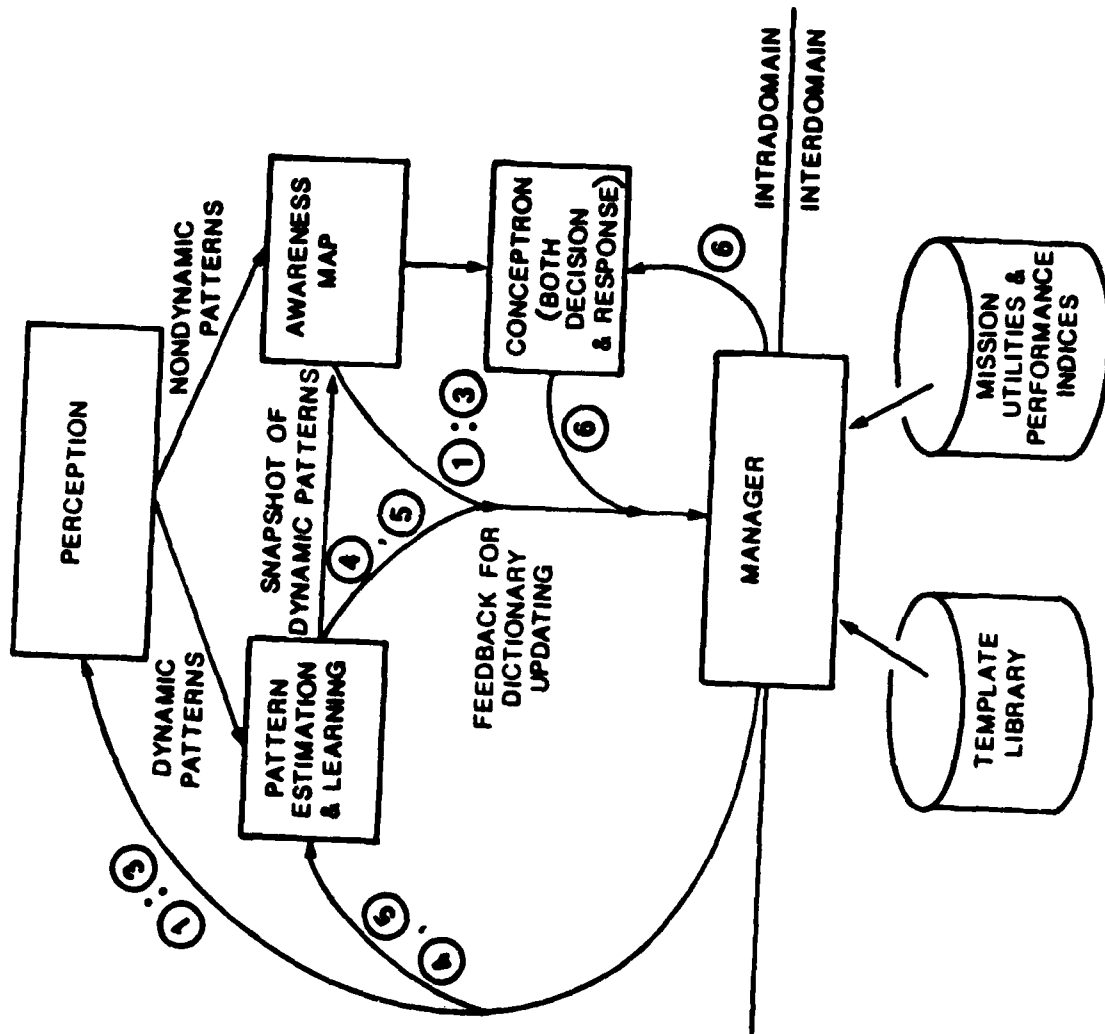


Figure 15. INTRADOMAIN USE OF DICTIONARIES (Refer to Figure 14 for numbers)

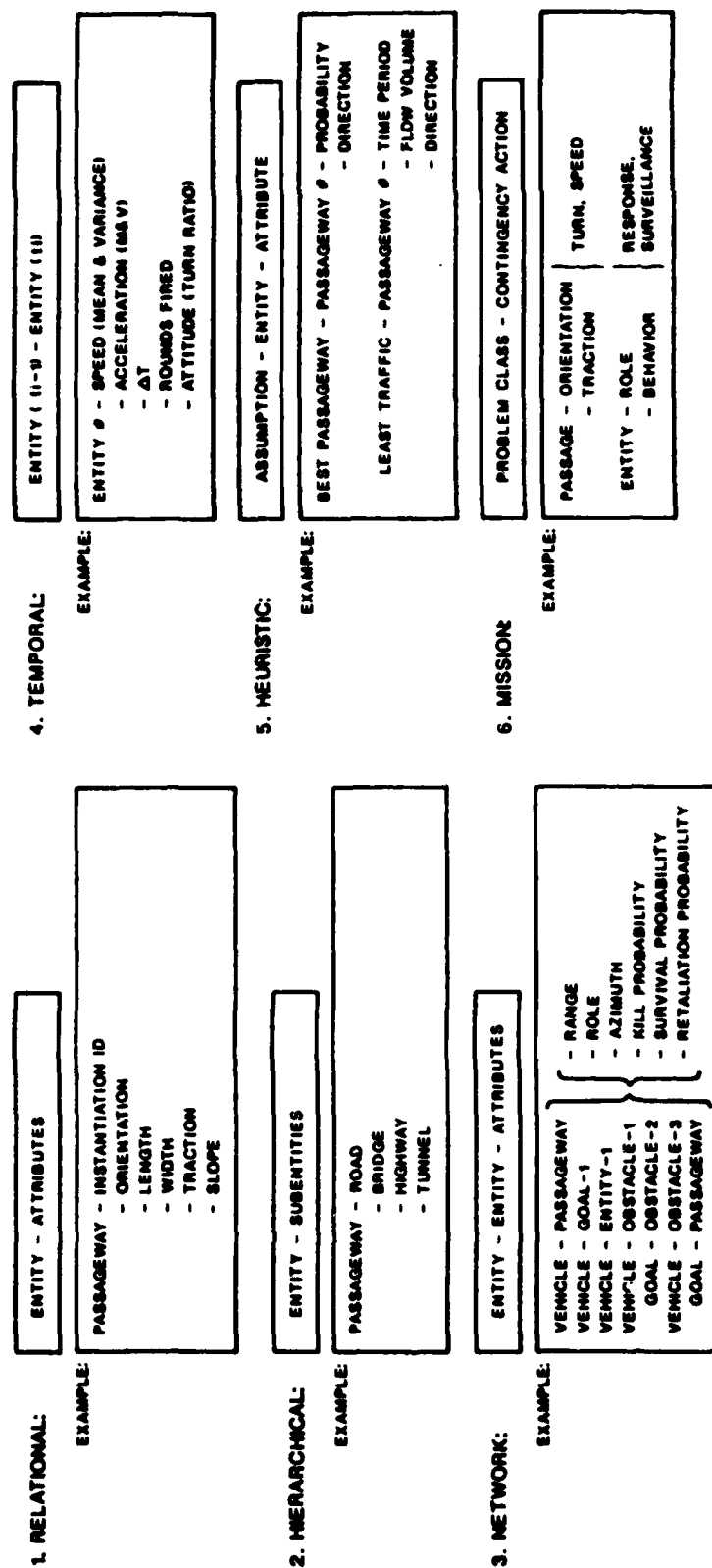


Figure 16. LIBRARY TEMPLATE COMPOSITION

The MANAGER preserves awareness integrity by logically linking the three views of a given instance, although, physically, these belong with their respective domain maps. This centralized mapping is supported by the dictionaries in the same manner a receptionist routes mail or calls. This way the vehicle's distributed view of the overall mission envelope, as centrally administrated by the MANAGER, is continuously updated.

Each domain perceives extrinsic information about the overall evolving mission envelope to the extent permissible by their respective sensors. In addition, the domains communicate status information about their own intrinsic performance and utility parameters critical to the overall vehicle survivability. This exchange among domains is also carried out by the MANAGER through its mapping dictionaries and protocol. The result is the vehicle's view of the mission envelope as a mosaic of the three subenvelopes.

A more effective and efficient architecture than that of a conventional expert system is thus obtained through an assemblage of distributed domains of expertise. Orthogonally to this partition, the subdomains of awareness, decision-making, and response are then allocated within the individual domains.

Mission envelope parameters, perceived linguistically and/or pictorially, are interpreted by a classifier which identifies an entity instance as either a dynamic or nondynamic type. Upon classification the instance's state is updated in the map half it belongs to. There is a map half for either type of entity. The reason behind this dichotomy in the domain maps is essentially functional more than logical. The separation or ranking of entities dictated by high speed real time attributes and nonstochastic instances relaxes the processing burden.

One of the halves in a given map reflects evolving situations posed by dynamic entities such as targets, threats or even other vehicles. Due to the evolving spatiotemporal nature of the mission envelope, and keeping present that there are mission goals to be accomplished within some performance and time constraints, particular and critical attention is paid by the PLANNER to the forecasting of future mission states. The PLANNER is to develop both mission goal schedules and contingency strategies in response to this (9,10).

Concurrently, and in the other half of the map, nondynamic scenic features such as terrain elements are interpreted with the lowest priority of temporal concern, but with a higher degree of "accountability leverage" (confidence) on which to base decisions and responses.

The three map dictionaries indexing the mission envelope are concurrently compiled and updated by the MANAGER as the mission evolves. These represent first-time and updated interdomain mappings linking multiple views of mission instances as perceived and classified through the sensors.

A given entity realization, in a given half of a given map, is jointly instantiated following realistic linguistic- and pictorial-type models supplied by ideal symbol and surface templates, respectively. The overall perception system is driven by a library of these entity (perceptual) tem-

plates containing selected attributes against which real entity instances are to be matched and classified (Figures 17:19). This bicameral (linguistical-pictorial) awareness approach is transparent to the MANAGER.

However, in following the "appropriateness" doctrine, how much or less redundancy these templates must bear without sacrificing certainty? There must be a compromise between what and how much is essential for the effective overall awareness of the mission scenario, and the associated storage and processing requirements. This reality must be kept in mind during the design of the different domains of expertise.

Perceptual templates are part of each entity's classification and are formulated for both dynamic and nondynamic entities. A priori (if so provided) and en route instances are described in conformity to these templates and stored in their respective maps. It must be borne with that not all features in a given template may be accounted for during the mission, in which case exception handling (DBMT terminology) must be provided for by the MANAGER.

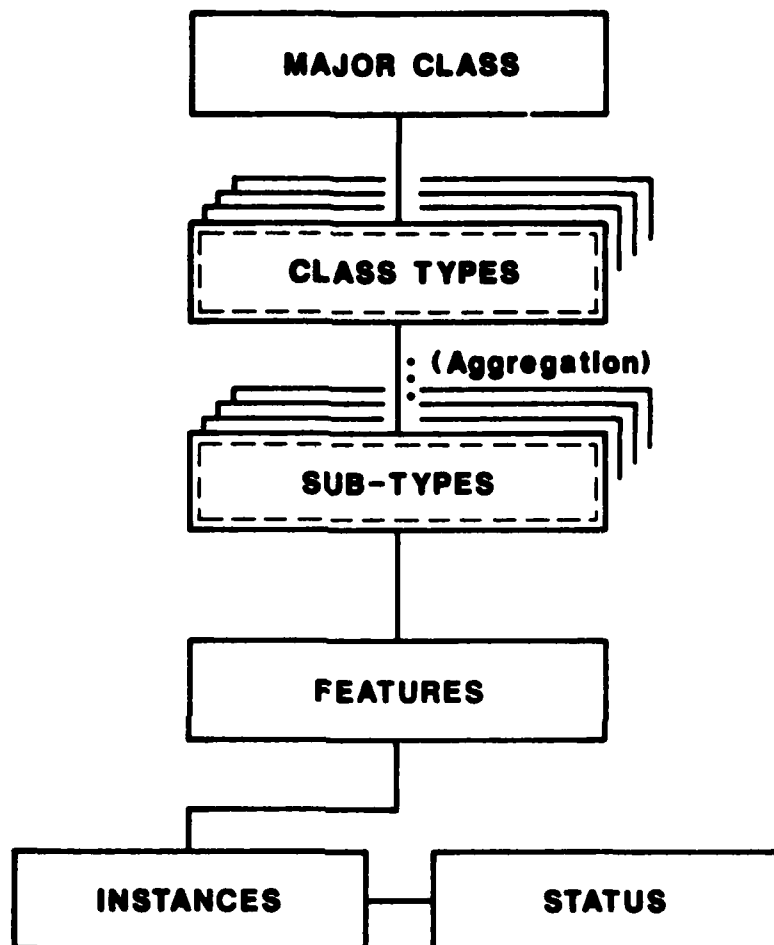


Figure 17. LOGICAL TEMPLATE FOR ENVIRONMENT ENTITIES

MAJOR CLASSES : 1. Topographic
 2. Weather
 3. Man-made Entities
 4. Mission Goals

CLASS TYPES : 1. Topographic : 1.1. Terrain
 1.2. Biota
 1.3. Waterways

SUB-TYPES : 1.1. Terrain : 1.1.1. Mountain
 1.1.2. Valley
 1.1.3. Crater
 1.2. Biota : 1.2.1. Forest
 1.2.2. Grass Field
 1.2.3. Crop Field
 1.3. Waterways : 1.3.1. River
 1.3.2. Lake
 1.3.3. Swamp

FEATURES : 1.1.1. Mountain : 1.1.1.1. Base Width
 1.1.1.2. Height
 1.1.1.3. Surface Condition
 1.1.1.4. Average Slope
 1.2.1. Forest : 1.2.1.1. Average Tree Height
 1.2.1.2. Average Stem Spacing
 1.2.1.3. Undergrowth Type
 1.3.1. River : 1.3.1.1. Average Depth
 1.3.1.2. Average Water Speed
 1.3.1.3. Average Bank Height
 1.3.1.4. Bottom Composition

(x.x.x.x.: Addressing Code For Objects And Attributes In The Templates)

Figure 18. SAMPLE FOUR CLASS TEMPLATE

TOPOGRAPHIC.WATERWAY.RIVER	Bottom Material	: Sand-Gravel
	Bottom Slope	: 5 degrees
	Ave Bank Height	: Level
	Ave Bank Slope	: 3 degrees
	Ave Depth	: 5 feet
	Ave Water Speed	: 10 mi/hr
	Ave Gap Width	: 200 feet

a. NONDYNAMIC INSTANTIATION

MAN-MADE.MOBILE.ARMORED	Vehicle Type	: Tank
	Shape	: Block
	Weight	: 15000 lbs
	Load	: 5000 lbs
	Composition	: Rugged Steel
	Max. Speed	: 40 mi/hr
	Max. Fire Power	: 90 mm cannon
	Max. Range	: 1000 yds
	Role Main	: Allied
	Kill Success	: .95
	Survivability	: .80
	Retaliation	: .90

(STATUS)	Speed	: 35 mi/hr
	Range	: 1500 yds.
	Heading	: 105 degrees
	Rounds Left	: 7
	Mission	: Aggressor
	Location	: Grid X ₁ Y ₁
	Est. Location	: Grid X ₄ Y ₆

b. DYNAMIC INSTANTIATION

Figure 19. INSTANTIATION SAMPLES

3.0 REQUIREMENTS IN SUPPORT OF DECISION-MAKING AND RESPONSE

3.1 The Generalized Conceptron

The vehicle's bicameral (i.e., linguistic and pictorial) approach lends itself to the distributed assemblage of maps. The proposed decision-making and response expert subdomains are based on a further modified Conceptron scheme applicable to either type of information, linguistic or pictorial. This scheme gradually generalizes (i.e., fuzzifies) an n -class entity map, by means of a pyramidal set of both deterministic and heuristic morphisms, into a reduced m -class map where $m < n$.

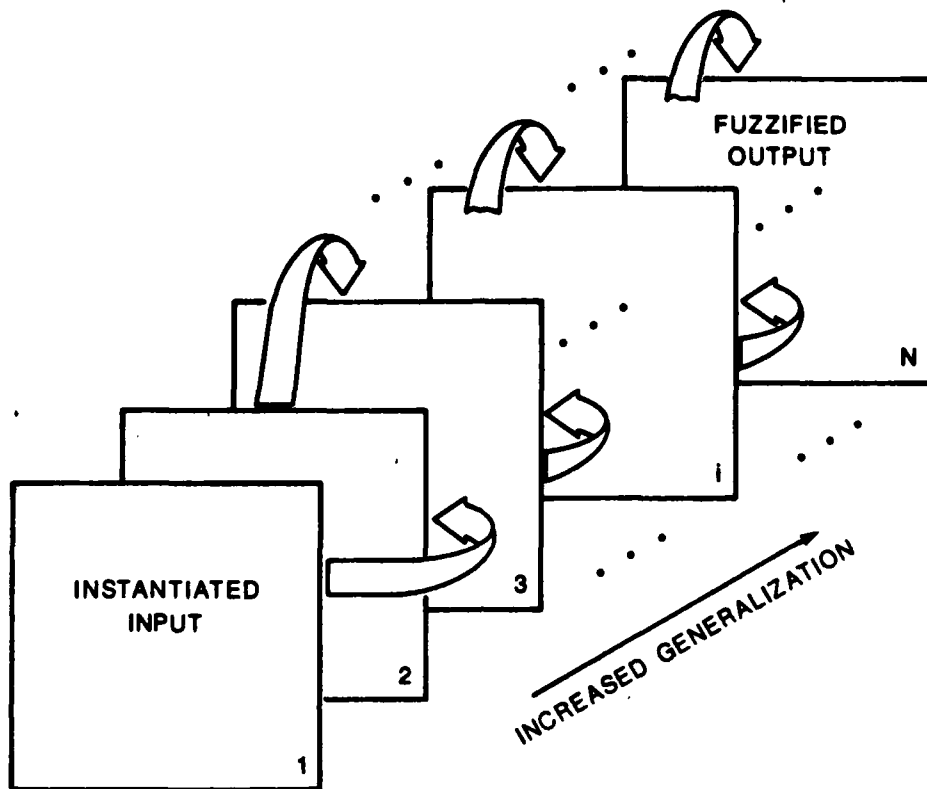
The concept of pyramidal decision-making and response is based on progressively fuzzy transformations along contiguous Conceptron-type planes (Figures 20,21). Each plane is defined by class sets of fuzzy functors orthogonally referenced by input and output axes. A given plane's input axis is mapped in from the output axes belonging to other planes. A plane has one output axis which may map several planes input axes. In this context there is a similarity with open-loop Perceptron units (11). In fact, an array of Perceptrons may be tabularly represented with a Conceptron model.

The inputs to the functors in the base plane of the eliciting decision-making Conceptron are the ordinary nonfuzzy instances stored in the mission maps. The latter are transformed onto fuzzy instances, in progressive fashion, throughout the rest of the pyramid, until a predetermined minimal number of causal categories (12) can be classified for response purposes (cause-effect principle). These categories become then the inputs to the base plane of the response Conceptron.

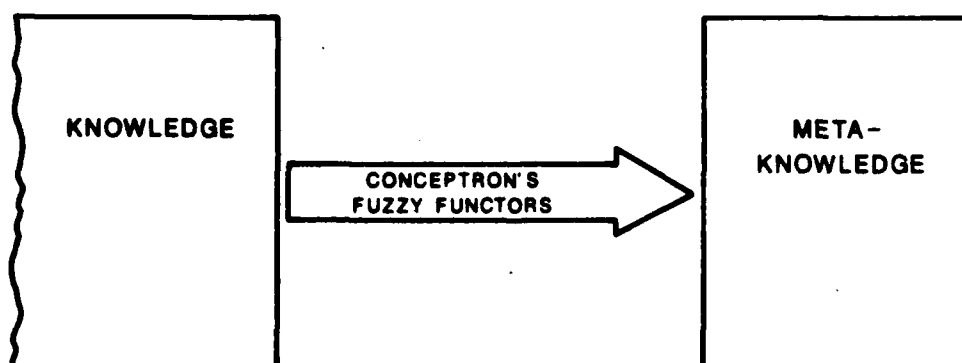
Depending on the entity features required at that level by design, the response Conceptron's plane inputs may be directly indexed by instances in the mission maps. The fuzzy transformation process is similarly executed in the response Conceptron's pyramid until the response outputs are in the form of action commands executable by the PILOT's actuators in the vehicle.

The vehicle should be able to respond without program modification to perturbed or somewhat inexactly predetermined situations. Fuzzy set theory supports this requirement by allowing linguistic biasing (13). For example, preferability, utility, or desirability parameters which are not easily and necessarily described in quantitative form are better handled if in qualitative form. In this context, fuzzy labels are used for all category instantiation, each with an associated fuzzy membership rank.

The vehicle's triad of cooperating domains (PLANNER, NAVIGATOR, PILOT) exchanges information, as pertinent, at the interdomain level. This also suggests the applicability of a Conceptron pyramid made up of logically contiguous domains of expertise. However, in this case the MANAGER acts as intermediary among interdomain planes by mapping outputs to inputs for the sake of integrity in the mission maps. These mappings link higher with lower level tuples made up of any pair combination from awareness, decision-making and response domain entries, and, definitely, their morphisms transform instance accuracy and cardinality.



a. PYRAMIDAL CONCEPTRON



b. FUZZY TRANSFORMATION

Figure 20. CONCEPTRON CONCEPT

First Functor Plane's Fuzzy Transforms:

- a) IF (SPEED > 55) THEN TOO FAST
 < 55 FAST
 < 35 AVERAGE
 < 15 SLOW
 < 5 NIL
- b) IF (TANK.LAPSED-SPARKS) THEN TANK.LAUNCH-BURSTS
- c) IF (ACOUSTIC SIGNAL POWER = 110 db) THEN LOUD
- d) IF (ACOUSTIC SIGNAL FREQ = 300 hz) THEN LOW PITCH
- e) IF (HEADING > 337.5 degrees AND < 22.5 degrees) THEN EAST

Second Functor Plane's Fuzzy Transforms:

- a) IF ((ACOUSTIC SIGNAL = LOUD) AND (ACOUSTIC SIGNAL = LOW PITCH)
 AND (TANK.LAUNCH-BURSTS)) THEN TANK.FIRING
- b) IF ((RELATIVE HEADING = WEST) AND (RANGE = CLOSE) AND
 (SPEED = FAST)) THEN POSSIBLE HEAD-ON COLLISION

Third Functor Plane's Fuzzy Transform:

IF ((TANK.FIRING) AND (RANGE = NEAR)) THEN HIGH-RISK ALERT

Fourth Functor Plane's Fuzzy Transform:

IF ((HIGH-RISK ALERT) AND (ROUNDS-LEFT = NIL)) THEN LOW-TRAVERSABILITY

Figure 21. SAMPLE INFERENTIAL CHAIN

To illustrate the Conceptron pyramid concept a simulation may consider a dual class transformation (e.g., speed or range) with perceived instantiations as input attributes. These are generalized by means of category transformation and interpreted as new attributes (i.e., fast or near) at the next functor plane in the pyramid. This next plane, in turn, may combine this fuzzy input with the fuzzy output from other different planes in order to form a new category (e.g., attitude of an entity relative to the vehicle) which in turn outputs further fuzzified attributes (e.g., threat, collision, evasion, etc).

As an aside, the generalization suggested here by the pyramidal approach is clearly different from the generalization supported by the "GEM" data base management system, from a DBMT point of view. The latter performs a series of "joins" resulting in expanded tables still carrying the original table "domains." In the pyramidal type of generalization these original categories are virtual. That is, they are not carried along into other functor planes. The object-driven pyramid fills in new categories (e.g., by induction or deduction) as a result of the evolution of entity classes in the vehicle's "brainstorming" process.

Fuzzy pyramidal generalization by functor planes involves the transformation of the most recently instanced map information, at one extreme of the pyramid, onto simplified and straightforward decisions and action commands, at the other. Between these extremes, generalization entails the normalization or clustering of the initial map instances into interim fuzzy categories. In turn, the latter are then unnormalized into finite attributes at the output, this time as instances of a new set of classes. Thus, at a given plane of transformation, and contrary to the DBMT methodology, entities are no longer treated by their initial finite peculiarities but by their relative membership (clustering) within fuzzy categories.

3.2 Applied Decision-Making and Response

The decision-making Conceptron in the PLANNER is able to discern on an instantiated strategic map (Figure 22) to produce a generalized binary one (Figure 23) on which to base its decisions. An interim map shows the ranking of the generalized traversability belonging to a given spatial grid in the strategic map (Figure 24). It points out, by means of admissibility thresholds, those areas potentially traversable by the vehicle.

In addition, the output of the tracing module (NAVIGATOR's decision-making Conceptron for guidance) deals with dynamic entities demanding stochastic surveying (Figure 25) and provides to the PLANNER a forecast (Figure 26) of future mid-course obstacles and hazards based on observed criteria. The dynamic feature map is generalized by the NAVIGATOR's decision-making Conceptron into an "estimated" map which complements the nondynamic one onto an overall composite map depicting the "appropriateness" of traversable passage-way alternatives within the vehicle's tactical and strategic ranges (Figure 27).

The continuous supply of information on both nondynamic and dynamic entities supports corrective attitude control and mission countermeasures. Again, a balance must exist between mission reachability (actual gross accomplishment of mission goals) and mission fidelity (accomplishment of mission goals as

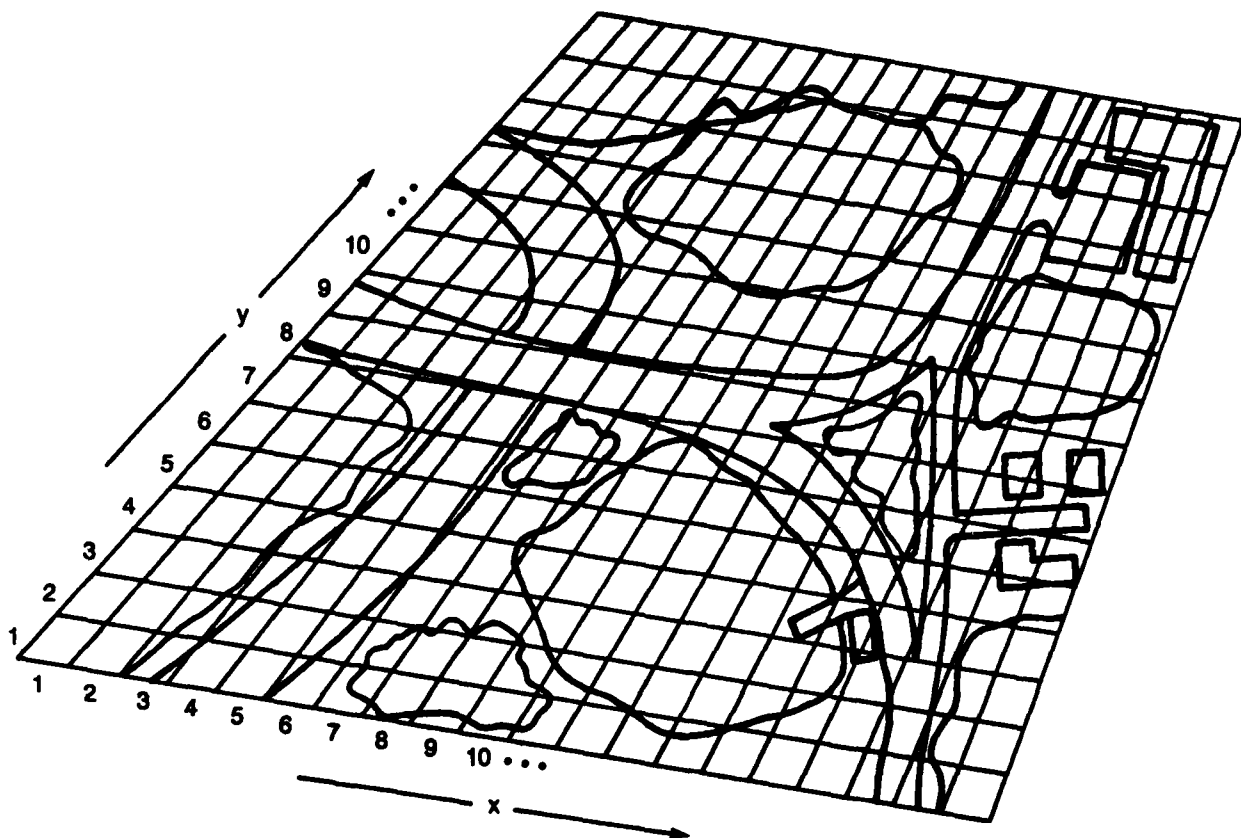


Figure 22. SAMPLE MISSION SCENARIO: CARTESIAN GRID OVERLAY

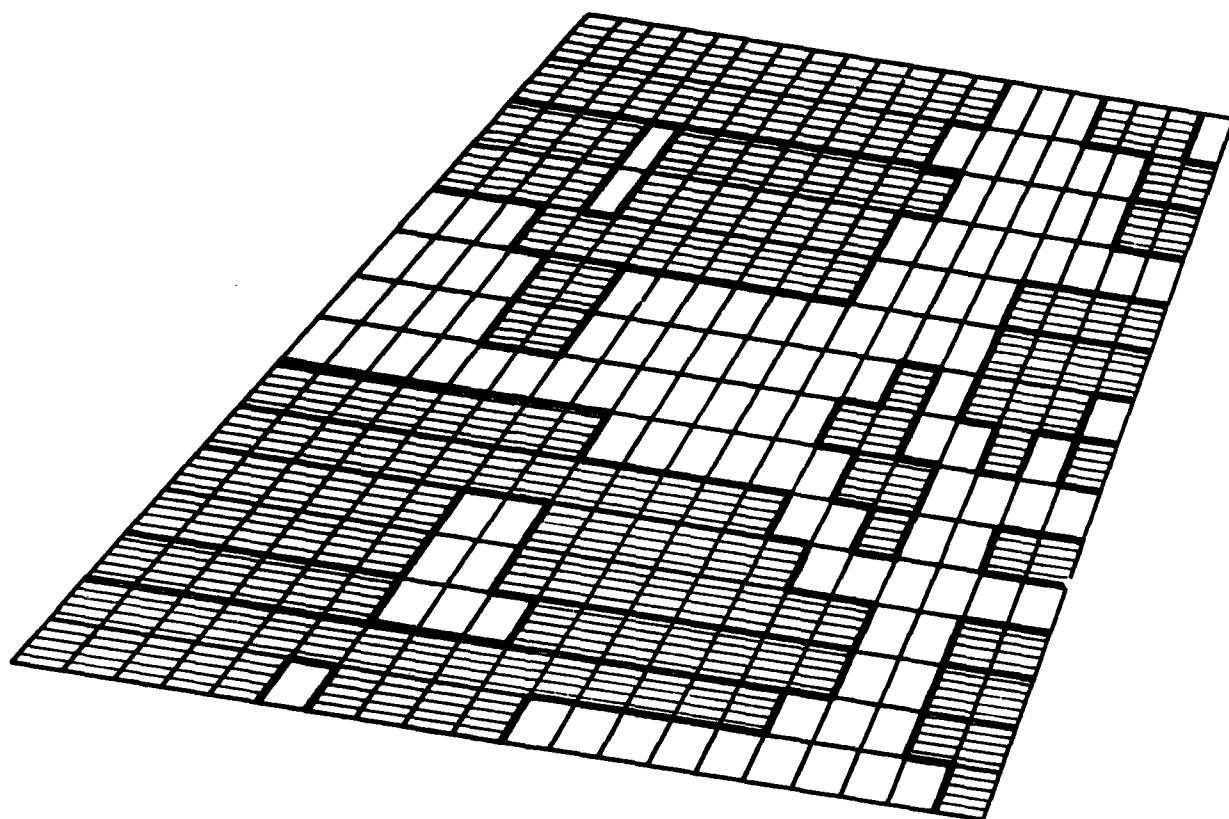
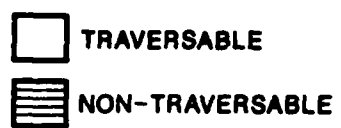


Figure 23. NONDYNAMIC TRAVERSABILITY MAP (BINARY)

FUZZY
 TRAVERSABILITY
 MEMBERSHIP

10
 EXCELLENT
 VERY GOOD
 GOOD
 :
 FAIR
 :
 POOR
 VERY POOR
 WORST
 0

STABLE MAP

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	6	10	10	0	0	0	6
1	1	1	1	1	1	4	4	3	3	3	3	6	7	10	10	10	10	0	0		
0	0	0	0	6	4	2	2	1	0	1	3	4	8	10	10	10	10	0	0		
0	0	0	0	6	4	2	2	2	2	2	3	7	8	10	10	10	10	6	6		
6	6	6	0	0	0	4	4	4	4	4	4	7	8	10	10	1	1	1	1		
6	6	6	6	0	0	6	6	6	6	6	6	7	8	10	10	1	1	1	1		
6	6	6	6	0	0	7	7	7	7	8	8	8	10	1	10	1	1	1	1	6	
10	10	10	10	10	10	10	10	10	10	10	10	10	10	1	1	10	10	0	6	0	
1	1	3	3	0	0	0	1	10	10	10	10	10	10	4	4	10	10	10	10		
1	2	3	0	0	0	2	2	3	0	0	0	0	10	10	4	10	10	0	0		
1	3	3	0	0	0	6	7	0	0	0	0	0	0	10	10	10	10	10	10		
1	3	0	0	0	0	6	7	0	0	0	0	0	0	0	10	10	3	3			
1	3	0	0	0	0	7	6	6	0	0	0	0	0	0	10	10	3	2			
1	3	0	0	0	1	3	3	3	1	2	1	1	0	0	6	10	10	3	2		
1	3	0	0	0	6	3	3	3	1		6	6	6	6	6	10	10	6	3		

(THRESHOLD - 5)

Figure 24. PRELIMINARY GRID BY GRID TRAVERSABILITY

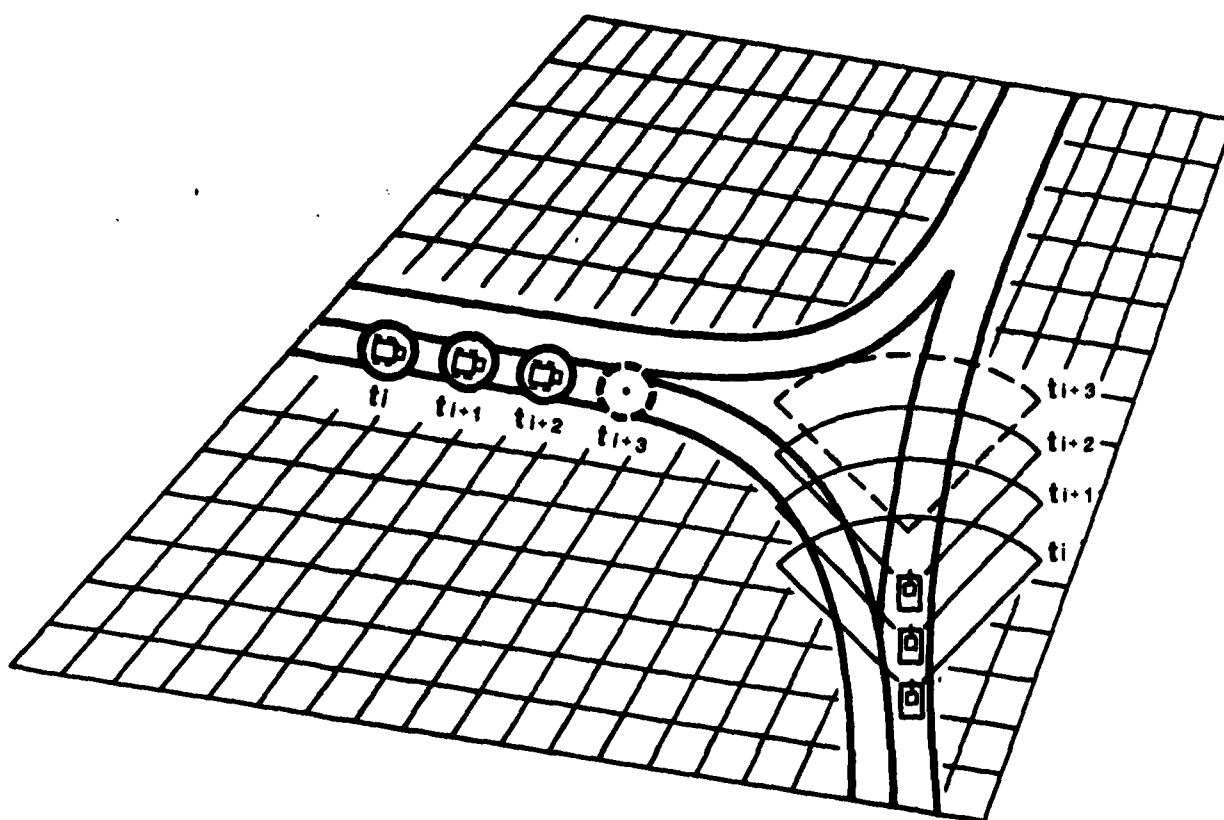


Figure 25. PRELIMINARY FORECASTING OF DYNAMIC ENTITIES

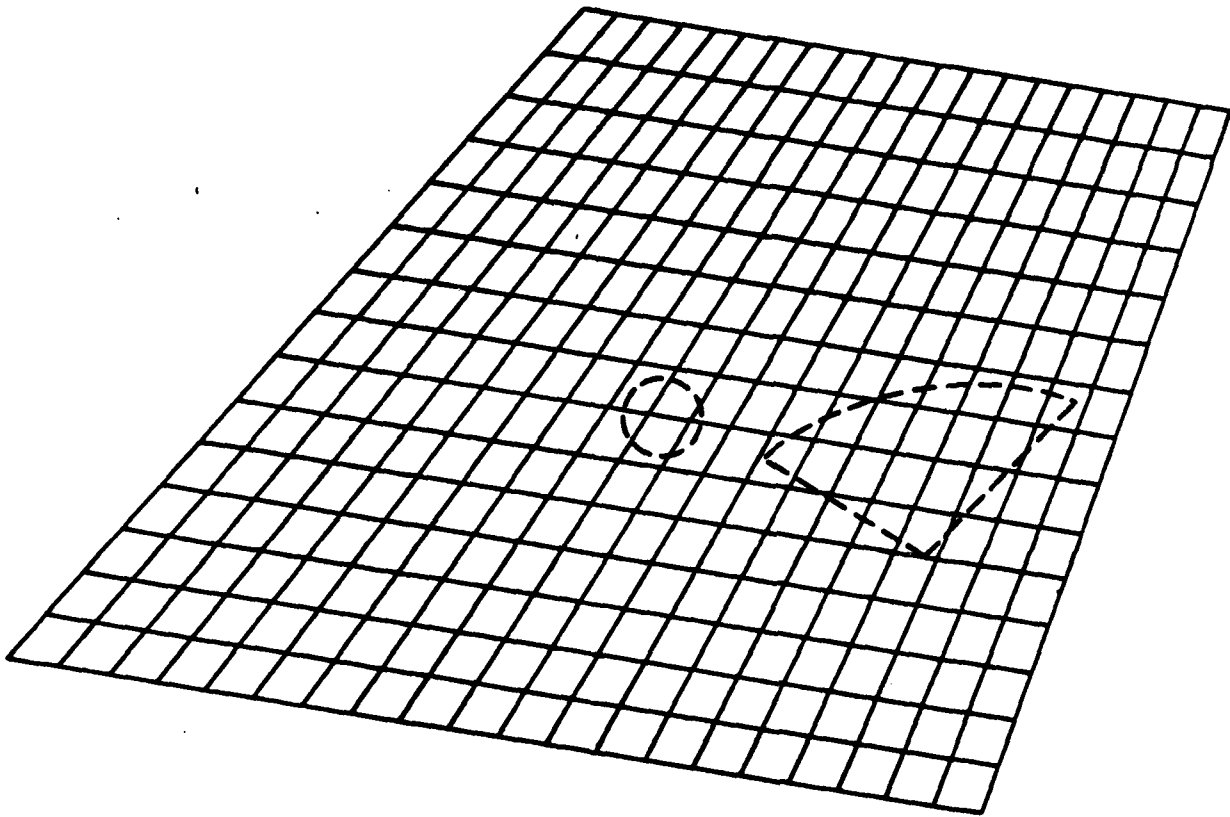


Figure 26. DYNAMIC MAP FORECASTED FOR t_1+3

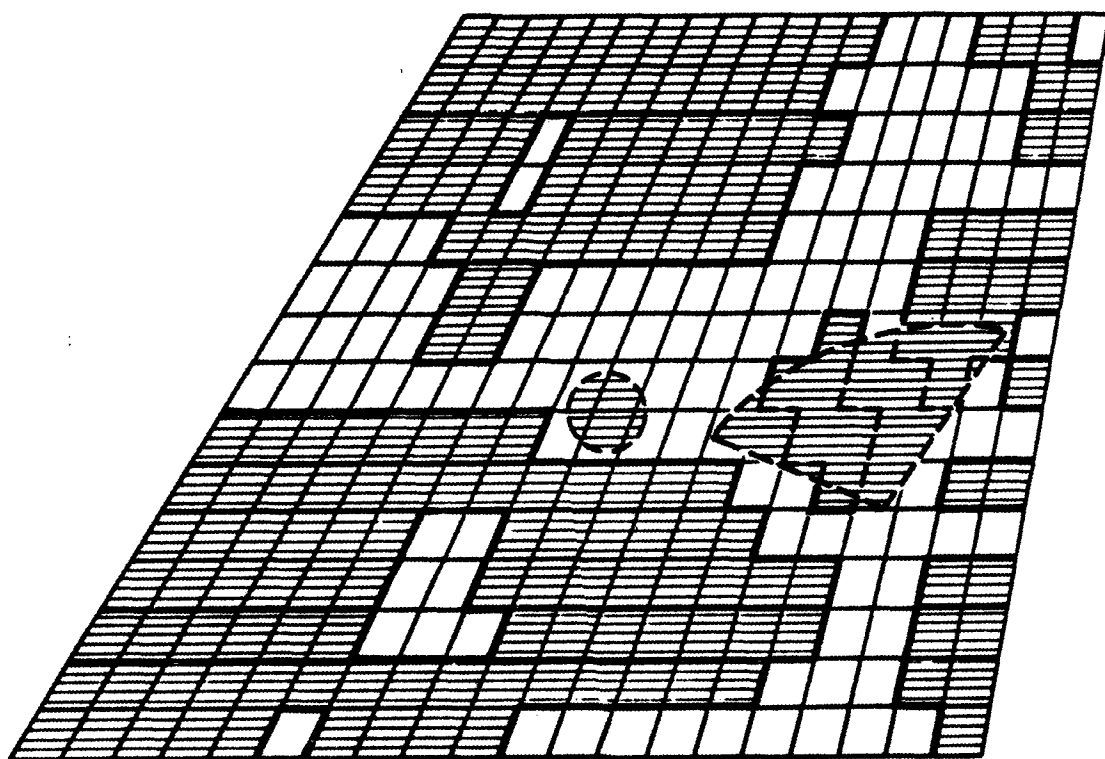


Figure 27. GENERALIZED TRAVERSABILITY MAP FOR 11x3

ideally scheduled). This is the concept established here as that of conformable guidance. The vehicle must choose between dealing with attitude control in order to follow the mission as strictly as possible, and, dealing with interim distracting obstacles or events.

Decision-making and response in the NAVIGATOR are considerably straightforward as far as their Conceptron implementation is concerned. However, both of the PLANNER's decision-making and response Conceptrons must deal with schedules and contingencies which are difficult to develop in a single pass. Assessment and planning are carried out using a recursive algorithm in which Conceptron planes must match current mission status information with desired or acceptable thresholds of performance. The Conceptron scheme is complemented by a modified A* search algorithm (1).

To illustrate the concept, the current mission state is represented by S_m and the arbitrary goal state by S_g , where S_m, S_g are members of S , and S is the state space in which S_m and S_g are possible. The tuple (S_m, S_g, S_v) is therefore understood as a state matrix representing actual, desired mean, and deviation entries for the mission status. The PLANNER's task is then to conform the S_m state vector to the S_g one, within allowable limits of the S_v vector. The subspace described by the S_g vector constitutes the collection of mission tasks and goals and is decomposable into a sequence of execution commands, in other words, implying a sequence of desired ideal mission states.

In this case, a Conceptron entry is expressed as a tuple of condition and action (C, I_w, A) , where I_w denotes a weighed implication (Figure 28). Each condition C is composed of a sequence of subconditions (C_1, C_2, \dots, C_i) which reflects its adaptability to recursive algorithms. An action is a mapping $Y(t) \rightarrow Y(t+T)$ between temporally contiguous mission states.

The function of the PILOT is to respond to the reality of the traversable path, and to control the actual vehicle motion. The response time of the PILOT should be minimal in order to compensate for the processing overhead induced by the initial uncertainty. The PILOT's mission map is rapidly updated by means of visual feedback. This quick response is achieved through:

- a. direct use of the frontal image (i.e., "windshield view") for motion control, instead of transforming it into a top view.

- b. assuming that the windshield view information consists of clear and obstructed areas which are defined under conditions of poor resolution (i.e., fuzzy boundaries). This also relaxes the image processing requirements to a realistic level. The MANAGER associates front and top views transparently and automatically.

The PILOT domain is based mainly on the most current visual information which is supplemented by the NAVIGATOR's responses. The PILOT's awareness on the mission scenario is prolonged temporarily only in contingency cases. Context abstraction is different for each domain of expertise's Conceptron. However, they all serve the purpose of object class reduction and increased rate of information processing. The PILOT's functors are selected such that the resulting motion elapses minimal time. The following guidelines are observed for that purpose:

PROBLEM-SOLUTION MATRIX ENTRIES

		TURN {			ACCELERATE			SPEED		
		NARROW			COAST					
			1	2	3	4	5	6		
DEVIATION {	SMALL	1	4	1	0	0	0	0	VISUAL PASSAGE FROM VEHICLE (SOLID LINES)	
	AVERAGE	2	1	3	0	0	0	0		
	LARGE	3	0	1	2	0	0	0		
TRAVERSABILITY {	EXCELLENT	4	10	10	10	2	1	0		
	FAIR	5	5	5	5	0	2	1		
	POOR	6	0	0	0	0	1	2		
DEVIATION {	SMALL	7	6	3	0	0	0	0	IMAGINARY PASSAGE TO GOAL (DOTTED LINES)	
	AVERAGE	8	3	5	0	0	0	0		
	LARGE	9	0	3	4	0	0	0		
TRAVERSABILITY {	EXCELLENT	10	10	10	10	1	0	0		
	FAIR	11	5	5	5	0	1	0		
	POOR	12	0	0	0	0	0	1		
COMPOSITE RANGE {	CLOSE	13	8	4	0	0	0	1	COMPOSITE RANGE	
	NEAR	14	4	6	0	0	1	0		
	FAR	15	0	4	5	1	0	0		
TURN {	NARROW	1	0	0	0	2	1	0		
	WIDE	2	0	0	0	1	2	0		
	AROUND	3	0	0	0	0	1	2		
SPEED {	ACCELERATE	4	2	1	0	0	0	0		
	CRUISE	5	0	2	1	0	0	0		
	COAST	6	0	1	2	0	0	0		

SOLUTION-SOLUTION MATRIX ENTRIES

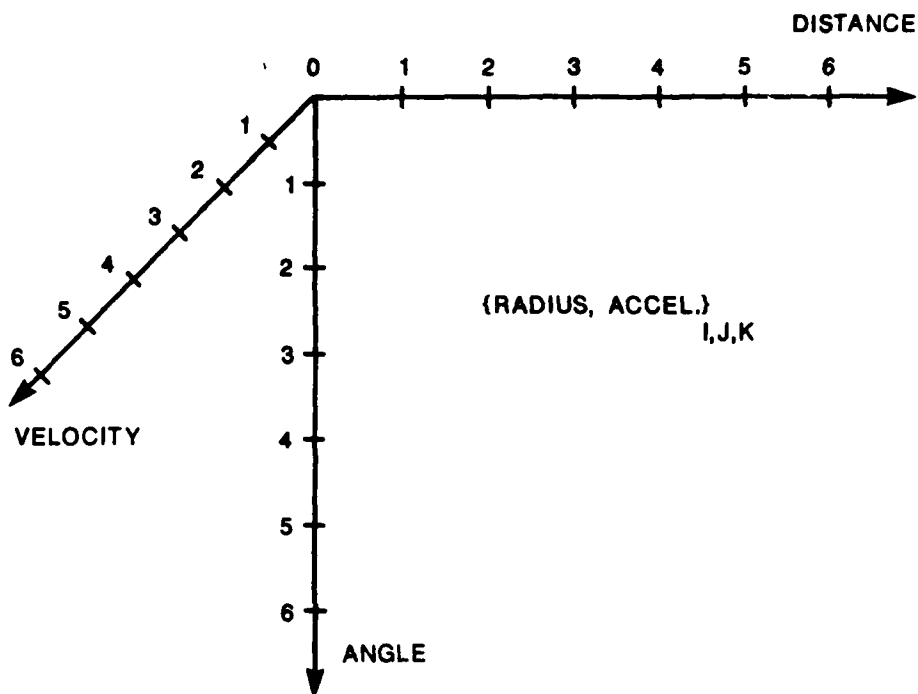
Figure 28. INITIAL CONCEPTRON ASSOCIATION WEIGHTS FOR NAVIGATOR

- a. The vehicle's velocity imposes a constraint of minimal turn radius.
- b. Distance range and attitude angle yield the required change in direction, which in turn imposes another constraint of minimal turn radius in order not to overshoot.
- c. The strategy for minimal time steering control consists of executing the sharpest turn (minimal radius) subject to (a) and (b) above.
- d. Relative vehicle attitude drift is corrected after the mean vehicle-to-next-goal angle. Attitude drift relative to a passageway is additionally corrected by considering a safe clearance from obstacle boundaries.
- e. Distance range imposes a limit in the vehicle's velocity. The closer the goal or obstacle, the smaller the maximal allowable velocity.
- f. If the estimated change in attitude is less than half of the required change, then the vehicle's velocity is decreased to allow for a sharper turn in the next command cycle.
- g. It follows then that the locomotion strategy for minimum time control is to apply maximal acceleration or maximal braking in order to follow the allowable trajectory (at maximal velocity) determined by (e) and (f) above.

This rule set is adapted into the pyramidal Conceptron, and since the precision associated with linguistical variables is kept at low levels (Figure 29), the planes are compact and easily accessible (Figure 30). The equivalent logic of decision-making is easily understood if portrayed in pseudo Pascal notation (Figure 31).

SET OR MEMBER	MNEMONIC	DESCRIPTION
N_{p1}	GOAL = (GOAL.DIST, GOAL.ANG)	Location of a goal relative to the vehicle in terms of distance and heading
N_{p2}	LEFTPWE = (LPWE.DIST, LPWE.ANG)	Entrance to a passageway on the vehicle's left, in terms of distance and heading
N_{p3}	RIGHTPWE = (RPWE.DIST, RPWE.ANG)	Same as above but at vehicle's right
N_{p4}	DANGER = (DAN.DIST, DAN.ANG)	The corner of an obstacle which disturbs proper motion, in terms of distance and heading
N_{p5}	VEL	Vehicle's velocity
N_{p6}	ACC	Vehicle's acceleration
N_{p7}	SENSORANG	Vision system's attitude
N_{p8}	CLEARCONE	No obstacles in the vehicle's heading
N_{p9}	SAFEMOVE	No dangerously close obstacles
N_{p10}	RADLEFT	Left turn radius
N_{p11}	RADRIGHT	Right turn radius
N_{p12}	WARNNAV	Warning to the NAVIGATOR on the excessive deviation from the prescribed path
N_{p13}	LASTPWE	Memory variable for the last turn made
ANG	{ahead, slight R/L deviation front R/L side, R/L side, back R/L side, behind}	Heading
S	(N_{p2} , N_{p3} , N_{p4} , N_{p8} , N_{p9})	Possible mission environment descriptions
X	(N_{p5} , N_{p6} , N_{p7} , N_{p10} , N_{p11} , N_{p12} , N_{p13})	Possible PILOT states
Y_1	(N_{p6})	Output to locomotion controller
Y_2	(N_{p10} , N_{p11})	Output to steering controller
Y_3	(N_{p7})	Output to sensor controller
Y_4	(N_{p12})	Output to NAVIGATOR
G	(N_{p1})	Possible goal states

Figure 29. PILOTS VOCABULARY



EXAMPLE:

RADIUS [D=5, A=2, V=3] = 15m



**IF (D=MED FAR) AND (A=MED NARROW) AND (V=SLOW)
THEN TURN RADIUS=15**

Figure 30. MATRIX REPRESENTATION OF PILOT'S CONCEPTION

```

Initialize Variables;
While GOAL.DIST > 0 DO
BEGIN
  Image Interpretation1 ( CLEARCONE );
  IF NOT CLEARCONE
  THEN
  BEGIN
    Proximity Sensing ( SAFEMOVE, DANGER );
    IF NOT SAFEMOVE
    THEN {Unsafe}
      ACC:= -1;
      IF DAN.ANG < 0
      THEN {Danger on the left}
        RADRIGHT:= AVOIDRULE [ DAN.DIST, DAN.ANG, VEL ]
      ELSE {Danger on the right}
        RADLEFT:= AVOIDRULE [ DAN.DIST, DAN.ANG, VEL ];
      ELSE {Safe}
        Image Interpretation2 (LPWE, RPWE, CLEARCONE);
        IF NOT CLEARCONE
        THEN {PWE}
          Choose Appropriate ( LPWE or RPWE );
          ACC:= LOCORULE [ L(R)PWE.DIST, L(R)PWE.ANG, VEL ];
          IF L(R)PWE.ANG < 0
          THEN
            RADLEFT:= AVOIDRULE [ L(R)PWE.DIST, L(R)PWE.ANG, VEL ]
          ELSE
            RADRIGHT:= AVOIDRULE [ L(R)PWE.DIST, L(R)PWE.ANG, VEL ];
          END; {Not clearcone}
        IF CLEARCONE
        THEN
          ACC:= LOCORULE [ GOAL.DIST, GOAL.ANG, VEL ]
          IF GOAL.ANG < 0
          THEN
            RADLEFT:= AVOIDRULE [ GOAL.DIST, GOAL.ANG, VEL ]
          ELSE
            RADRIGHT:= AVOIDRULE [ GOAL.DIST, GOAL.ANG, VEL ];
          Find Sensor Angle (SENSORANG );
          Find Deviation from Path ( WARNNAV );
          Simulate Steering;
          Simulate Locomotion;
          Simulate Feedback;
        END; {while}
      END;
    END;
  END;

```

Figure 31. PSUEDO PASCAL VERSION OF PILOT CONCEPTION

4.0 AWARENESS THROUGH THE MISSION MAPS

4.1 Logical Design and Suggested Implementation

The design suggested in this paper for the knowledge base (mission maps) is nontraditional in the sense that it incorporates each of the DBMT-suggested database structures. This is so in order to:

- a. ease the burden posed by autonomous information management in real time (with hierarchical- and network-type architectures), and,
- b. facilitate programmer access for maintenance purposes (with a relational-type architecture, needless otherwise).

The logical design of the maps integrates hierarchical attribute lattices as subsets of a network of entity lattices. The former establish entity attribute instances as loaded or updated during the vehicle's mission, while the latter organize these entities into clusters relative to the vehicle's state in the mission. The networking is necessary for the tracing of dynamic entity instances within the multivariate (spatiotemporal-plus) framework of reference which places the vehicle at the center of the evolving mission envelope.

The utilization of hierarchical lattices for the representation of entity attributes becomes obvious after observing the following:

- a. Mission entities are decomposable into an inherent top/down (parent/-child) structure similar to the DBTG's set definition with more than one dependent level. This is also analogous to the frames technique developed by Minsky at MIT. The structure under consideration here is called a lattice, where each node represents a class attribute and each edge is a fuzzy relation between parent and child or child and child (sibling) attributes.
- b. A hierarchical lattice is best suited for representing multiple, nonuniform levels of aggregation, characteristic of polymorphic entities. However, level cardinality is prefixed by the template catalog, which limits the amount of children a given parent has, and forces one parent per children. This is the reason for not using a network approach throughout since each and every entity's attribute instances are unique and do not share their parent instance with other parental siblings. Also, hierarchical branching implies faster maintenance paths than network links do.
- c. Due to the circumstantial nature of dynamic entities, weak relations are expected (virtual, at least because of the spatiotemporal nature). Therefore, enforcement of entity and/or referential integrity is a must. Furthermore, because of b) above, DBMT approaches (e.g., relational tables) must not be sought since these force a sparse structure with multiple, complex subtypes. Normalization becomes impossible, and data manipulation becomes impractical, with too much overhead for either indexing or scanning, if either is ever attempted and afforded.

Another advantage of a hierarchical structure is that it conforms to the concept of pyramidal transformation by facilitating gradual abstraction

(fuzzification). A given fuzzy functor plane in the NAVIGATOR's decision-making Conceptron may produce its transformation by mapping-in a parent rather than a child instance. For example, the instance of a "lake" is sufficient information to determine local traversability by the NAVIGATOR, without proceeding further down the instance's hierarchical lattice to test for "depth", "bottom composition", etc, all of which pertain to the PILOT's decision-making Conceptron. This of course speeds up the vehicle's overall performance.

In order to provide adaptation to dynamic situations, a network is implemented which establishes the access paths for both the perception-to-mission-map and the mission-map-to-decision-making or -response mapping interfaces. The reasons for choosing a network approach are as follows:

a. Interentity relativity cannot be expressed with hierarchical lattices since these relations are not normalized in essence. What may be ranked as parent at one time may not remain as such at a later instant due to either real circumstantial dynamics, or to decision-making or response transformations through mission evolution. That is, entities become virtual. Virtual connectivity among entities, and sets of entities, is supported by networking.

b. The network approach inherently lends itself for the faithful-as-possible representation of the mission scenario and states. The vehicle's awareness of the mission's reality is a map portraying entities distributed within an imaginary pattern of uniform grids which conforms to the 3-dimensional topography of the terrain (Figures 32,22). The conversion from a given area of the mission scenario to a given grid screen in the mission map follows the Defense Mapping Agency's standards. A separate screen is imaginarily overlaid on the global mission map for each of the domain maps in the vehicle. A given domain map's grid screen conforms to a relative grid size corresponding to the ranges of perception, awareness, decision-making, and response belonging to the domain of expertise associated with the map (Figure 10).

c. Due to the circumstantial randomness of entity instantiation, an object-oriented random storing scheme is adopted. This facilitates updates, deletions and insertions as needed, in minimal time. Access is by means of links, with no need for sorting or ordering at all. Logically speaking, a given domain map may be viewed as a network collection of hierarchical lattices within imaginarily clustering spatial grids.

Spatiotemporal evolution, in this case, is treated with a "region or cluster of sets" ("set of sets") approach, enabling virtual addition and/or deletion of sets as the spatial range implied by a given domain's map "expands" or "contracts" while the vehicle imaginarily "wanders" through the global mission map (Figure 33). This temporally virtual qualification for sets is complemented and complicated by dynamic entities which are virtual themselves within a given set. That is, these entities are virtual twice, in a figure of speech. The utilization of forward and backward network links facilitates access for the real-time maintenance of these spatiotemporally dynamic types (Figure 34).

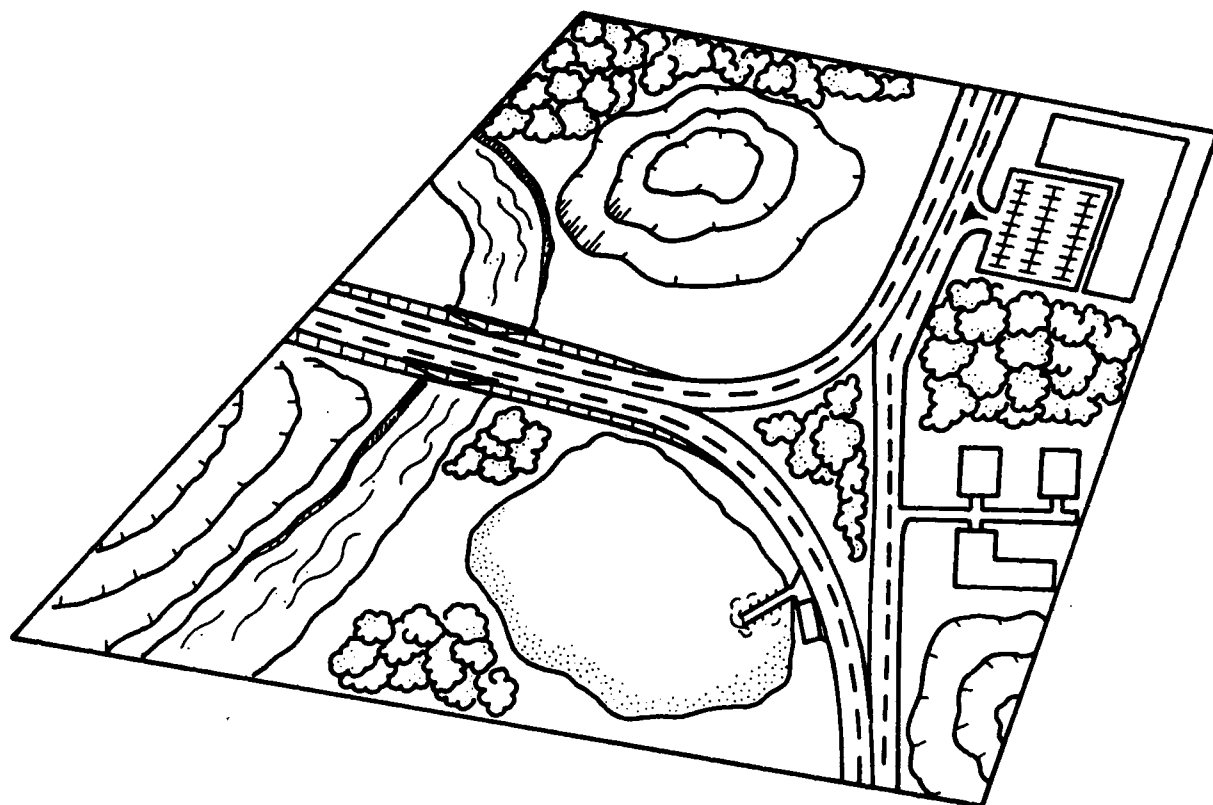
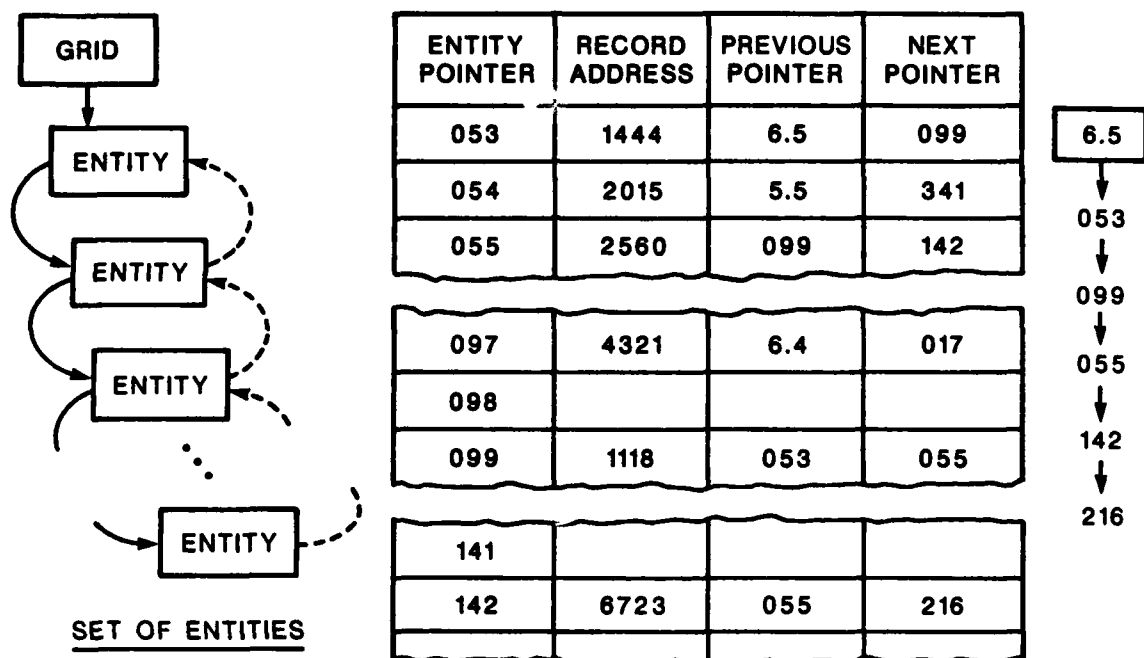
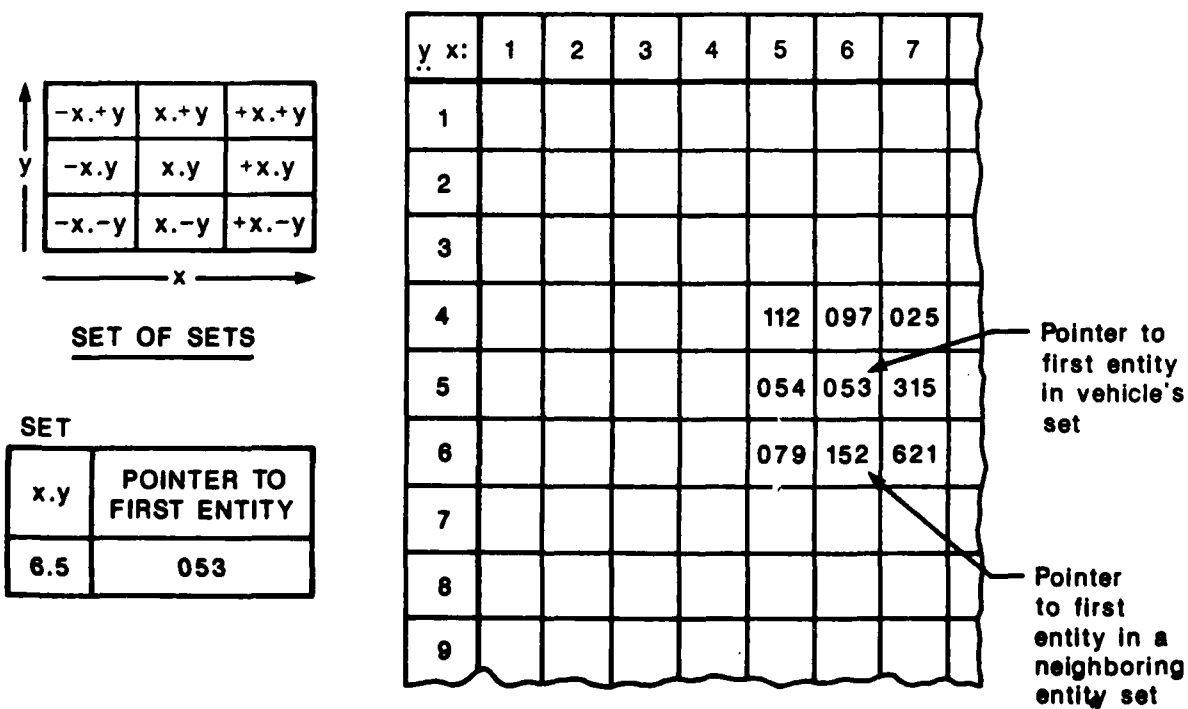


Figure 32. SAMPLE MISSION SCENARIO: ORIGINAL MAP



a. LINKED ENTITY INDEXING: RANDOM STORAGE



b. CARTESIAN CLUSTERING OF SETS

Figure 34. NETWORK SCHEME

The random storage also saves space by adjusting, or conforming, itself to the heterogeneous nonuniform type instantiation of entity lattices. Addition of new instances entails some kind of sequential random access similar to those in disk file storage units. These search for the first block with enough volume for the creation of a record. This suggests the inclusion of a "squeeze" type of maintenance utility able to temporarily compress the contents in memory whenever it seems there is not enough contiguous hardware volume to fit a particular instance's record. However, the need for this remains to be formally justified.

4.2 Access to the Information in the Mission Maps

The underlying data constructs and handling mechanisms proposed by the DBTG standards (8) are compatible with this design in their full scope. These become handy in the development and production cycles of a project which considers the design suggested here. That is, during the compilation, implementation, and maintenance of a database containing the mission maps and dictionaries.

Again, and because of the virtual and random membership, entities are best handled with "MANUAL insertion" and "OPTIONAL retention", allowing full utilization of DML (Data Manipulation Language) statements such as "CONNECT", "DISCONNECT" and "RECONNECT". "SET SELECTION" is performed by correlating perceived entity instances to the ideal ones represented in the lattice templates utilized by the perception subdomain, and then, linking the record to the clustered spatial set representing the real world grid where the real world entity is physically located. That is, instantiated entities may be connected, disconnected, and reconnected to/from set occurrences at will.

The "SET SELECTION" within the clustered spatial set is performed by applying an octal grid screen, conforming to the eight immediate grids adjacent to the vehicle's own (fuzzy spatial proximity). That is, sets are clustered by spatial connectivity to the vehicle's immediate surroundings, also at will. Of course, the size of this screen is arbitrary and depends on the vehicle's speed, and/or the range of decision-making and response for a particular domain of expertise (Figure 10). Due to the logical arrangement of the grids in an imaginary 2-dimensional cartesian plane, clustering into a screen is indexed by a relative coordinate pair.

At the output of a given transformation pyramid, entity instances and the sets of entities may be accessed through relational tables to facilitate maintenance by the expert system designer (Figure 35). The corresponding fuzzified domains (categories) and attributes are normalized across all instances providing this way a scale for measuring preferability, utility, desirability, et al. If the vehicle is to be simulated, this relational table approach becomes the main monitoring tool.

Several aspects of DBMT theory which relate to the proposed design may be discussed briefly:

- a. There is no mission map sharing in the sense of multiuser access. The MANAGER's indexing scheme is used to locate interdomain instances and supervise the communication protocol. The main function of the decision-making and response Conceptor is to execute decisions and actions as prompted by the information premapped from the mission maps. However, where necessary,

SPATIAL SETS	ID	TRAVERSABILITY	ENTITY	GOAL	THREATS
PLANNER	5 5	POOR	DYNAMIC	NEAR	ARTILLERY
	5.6	FAIR	MIXED	CLOSE	MINE FIELD
	5.7	EXCELLENT	STABLE	CLOSE	NIL
	6.5	GOOD	MIXED	FAR	SMALL CALIBER

ENTITY	ID	SET	CLASS	ROLE	ATTITUDE	RANGE	AZIM
NAVIGATOR	053	5.5	DYNAMIC	ENEMY	FIRING	NEAR	FRONT
	091	5 5	DYNAMIC	GOAL	EVADING	CLOSE	L.SIDE
	140	5.6	STABLE	ENEMY	STAND-BY	FAR	F.L.SIDE
	215	5.7	STABLE	ALLIED	SUPPORT	NEXT	R.SIDE

ENTITY/SET	AZIM	CONDITION	GOAL	VEHICLE	PROJECTED
PILOT	S	BUMPY	AHEAD	ATTACK	EARLY
	SW	SMOOTH	DEV.	ATTACK	ON TIME
	W	SMOOTH	DEV.	COVER	ON TIME
	NW	SMOOTH	AWAY	COVER	LATE
	N	SLIPPERY	AWAY	COVER	TOO LATE

Figure 35. SAMPLE RELATIONAL TABLES

the functor planes may read from the maps in order to additionally perform transformations on the information and write back into any of the maps, if the output warrant (e.g., when updating parameter status information). In the case of interdomain "write" transactions these are also arbitrated by the MANAGER. Therefore, there is an inherent logical insulation between maps which eases the administration of the distributed map architecture presented here.

Map access privileges must be supervised by the MANAGER. Transaction lengths may be arbitrated with "locks" as in DBMT-based implementations, allowing retrieval of clusters of sets in one sweep. Better yet, memory could be partitioned in a "flip-flop" fashion, with "reads" and "writes" to alternating locations. "Reading" taking place in the most recently updated half and "writing" in the opposite one.

b. Both information integrity and precedence are enforced because of the centralized acquisition and distribution scheme. Nevertheless, some kind of pointer scheme is necessary in order to establish a continuity link once nonfuzzy instances are mapped onto either fuzzified relational tables (for maintenance), or other domains transformation pyramids. The MANAGER's dictionary serves this purpose. It facilitates means for locating and performing DML operations on fuzzy tuples (as performed in System R (8)). Map access may be initiated when a nonfuzzy instance have induced a fuzzy update due to circumstances affecting its generalization, for example.

It should be obvious by now that the suggested design of the mission maps is supported by the three main DBMT architectural forms (with their respective access schemes): hierarchical lattices embedded in network sets, plus, relational tables. As mentioned before, this is so in order to make the vehicle an overall object-driven autonomous expert system, thus modeling the mission maps after the natural structure of the real mission scenario and states. Instead of forcing the maps to a restrictive architecture by modeling the mission environment after a fixed scheme. The latter would mean a more expensive implementation, if realizable at all. The logical order inherent in this architectural collage allows "collision free" concurrent processing by the different domains of expertise. The MANAGER is just a name for the abstract scheme that results.

5.0 SIMULATION RESULTS

The vehicle's NAVIGATOR domain was simulated in a VAX 780 system using Pascal. Time-dependent performance was not stressed since the mission scenario was a nondynamic feature map. One of the objectives was to analyze the reasoning behind the decisions made by the vehicle while traversing an unknown terrain. It was assumed that the vehicle would recognize features once sensed. However, their existence was unknown a priori. Awareness was on-the-spot. The only landmarks provided were start and final locations.

The vehicle's NAVIGATOR was to constitute: a domain gathering, and processing sensorial signals (from the PERCEPTION subdomain), and synthesized strategic information (from the PLANNER domain); assessing the tactical situation; and, delegating maneuvering commands for motion-control (to the PILOT domain).

The experiment was limited to a NAVIGATOR producing steering and speeding commands taking the vehicle from start to final locations. A PLANNER was also simulated separately and with a different mission map for input. The simulations were aimed mainly at comparing two or more fuzzy input vocabularies and their effect in Conceptron's performance. In addition, the PLANNER's simulation tested contingent route planning induced by deliberately switching among fuzzy attributes at the input of its decision-making Conceptron plane.

The map features recognizable by the vehicle were provided with the following vocabulary: grass field, mud field, paved road, bridge, shallow river, mountain, forest, building, lake, wall, and start/finish landmarks (Figure 36) (instance size, shape, and location not supplied). The mission was to reach the final goal in minimal time, while providing commands for relative change in attitude (turn) and speed as output.

Originally, the elapsed mission time was judged on both attitude deviation from the final goal's landmark and terrain's traversability factor. Minimal attitude deviation implied the selection of passageways which better approximated a straight path to the final goal (efficient point-to-point navigation). The traversability factor considered the state of the passageways, which could have an effect in the vehicle's maneuvering performance. The visual window of the PILOT's sensor system was set at 45 degrees. Decisions to be made about situations within that window arc were delegated to the PILOT. The visual range was set at ten map grids maximum.

The input to the NAVIGATOR was based on a fuzzy description of the situation enclosed within its range of decision-making and response. Two fuzzy classes were considered: passageway deviation and traversability. Passageway orientation relative to the vehicle's heading was considered "small" deviation if its azimuth was in the (+/-) 45:90 degree arc, for example. "Average" and "large" deviations accounted for azimuths in the 90:135 and above-135 arcs, respectively.

A binary traversability (obstacle, nonobstacle) was assigned to the features in the map. The set of obstacles included: river, mountain, forest, building, lake, and wall. The nonobstacles were then ranked as excellent (road, bridge), or fair (grass field) or poor (mud, shallow river).

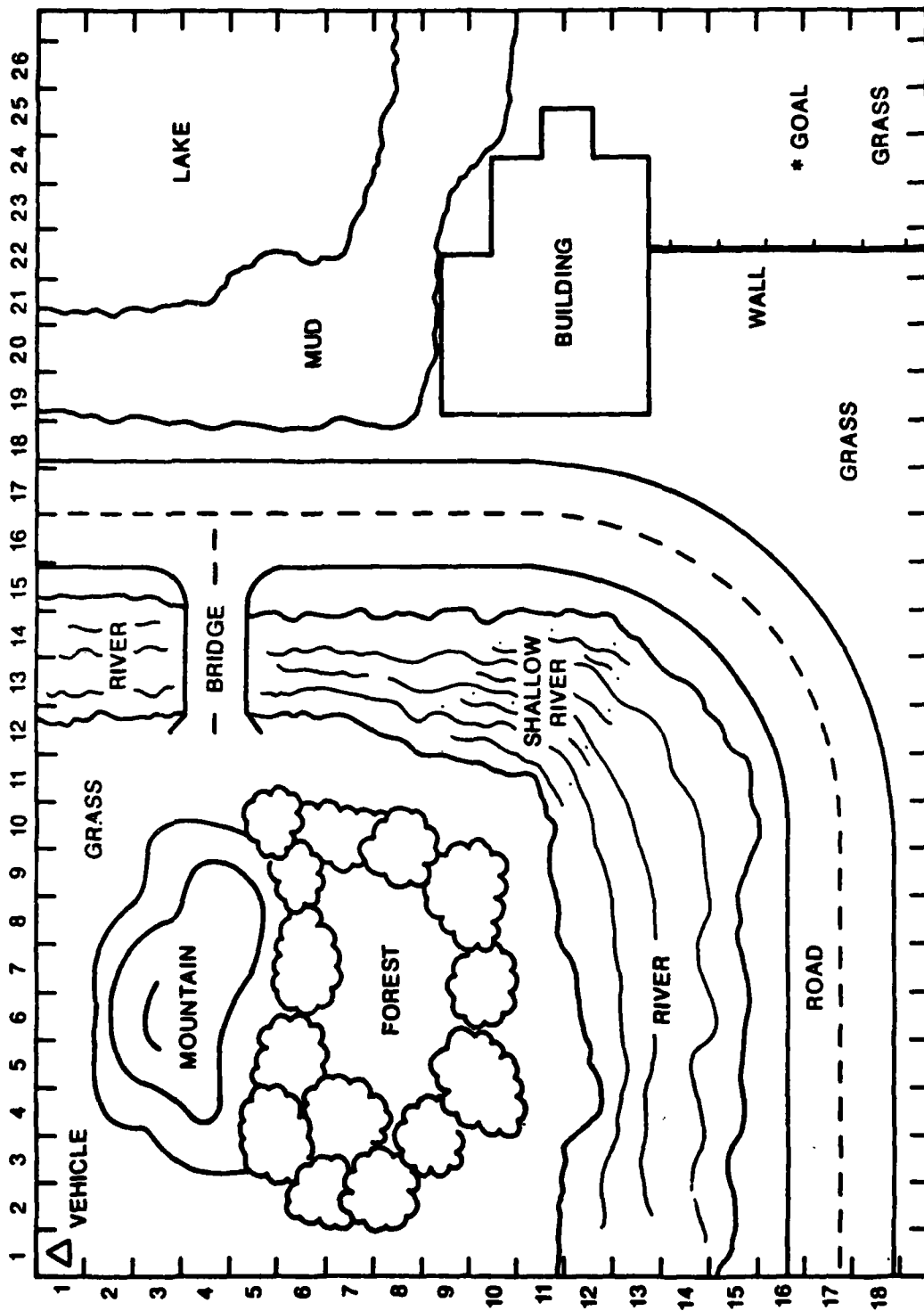


Figure 36. SIMULATION SCENARIO

Each imaginary grid in the map was assigned an average binary traversability due to the composite of entities present in the grid. Each passageway was then assigned an average traversability dependent on the imaginary grids composing it (Figures 37, 38).

The output from the NAVIGATOR specified commands to the PILOT in the form of two fuzzy classes: turn and speed. Logically, and provided a PILOT domain was available, these commands would have been supplemented and applied by the vehicle's actuators with a less fuzzier (more accurate) description of the terrain conditions and the vehicle's traversing capabilities, suitable to the PILOT's set of mission map, decision-making, and response Conceptiontrons.

The PILOT was commanded to make a "narrow" turn upon a passageway in the (+/-) 45:90 arc. "Wide" and "around" commands were issued for turn attitudes within the 90:135 and above-135 arcs, respectively. The PILOT was to "accelerate" if the difference between passageway length and range to the goal was greater than 10 unit map grids. "Cruise" and "coast" commands were issued for differences of 5:10 and below-5 unit map grids, respectively.

During simulation it was determined that extra information was necessary and thus the fuzzy vocabulary was expanded to include three more descriptive input classes. The situations that prompted this change were two. First, at some points of the mission, PILOT commands were ambiguous (e.g., left and right turns were equivalently graded due to geometric congruency among passageways). Instead of opting for an arbitrary command, heuristics were added to the NAVIGATOR's Conceptiontron as follows.

An imaginary passageway A' was to connect the exit of the passage A, under observation, with the final goal's landmark. Its average traversability was to be determined by any recognizable feature detected from the vehicle's location, or else, assumed equivalent to that of passageway A. Its deviation was determined relative to the vehicle-to-final-goal range line. This resulted in four input classes: passageway deviation and traversability for both A and A'.

A fifth input class was introduced due to a second situation. Mainly, the A-A' passageway combination did not necessarily imply overall minimal distance with a minimal composite deviation. Therefore, the fuzzy distance range of the A-A' combination was to be considered. In addition, a third output class was under consideration: turning rate, or angular speed. However, its inclusion was outruled since its objective was met with the logical integration of the turn-speed classes. That is, the functor weight entries for turn-speed were coordinated onto a metarelationship, resulting in a slower speed under a wider turn, for example. Inclusion of this third class would have resulted in redundancy.

It was noted that the higher the redundancy the easier to program the Conceptiontron. This implied more linearly dependent classes and an extended mapping dictionary for functor planes and pyramids. But resulted in a reduced burden on the programmer's expertise since subtle and not so obvious entity meta-interrelationships could not pass undetected. Given the simplicity of this particular experiment, complexity of the Conceptiontron planes was not a critical issue.

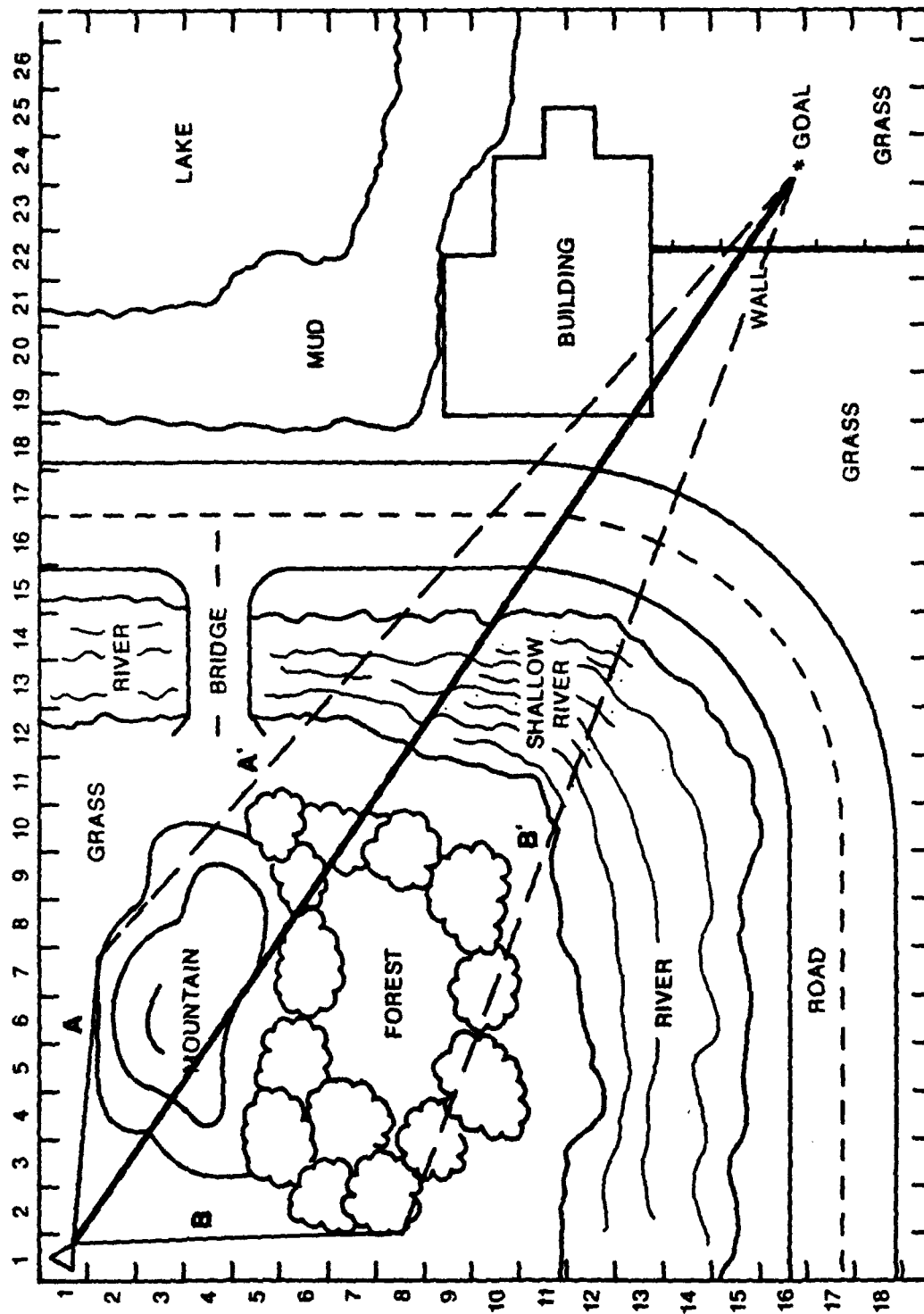
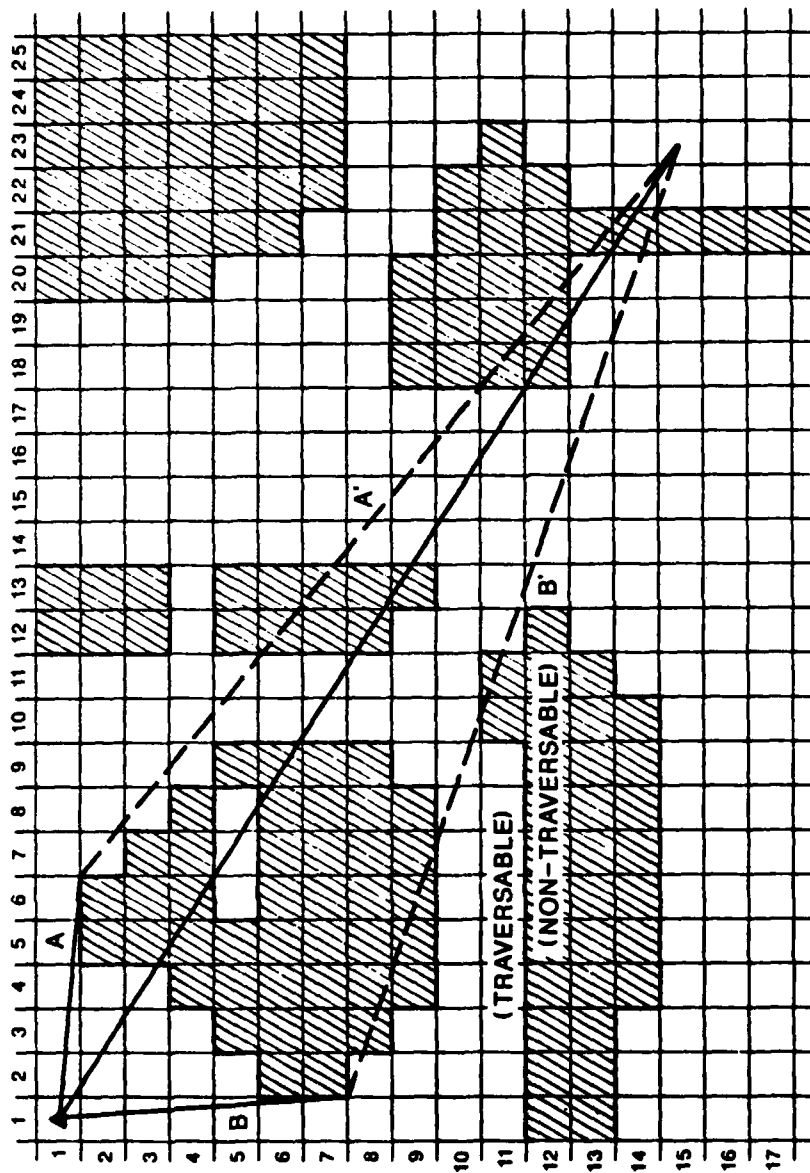


Figure 37. INITIAL MOVE



GOAL RANGE-52

AVG. TRAVERSABILITY:

A:5 A':5

A+A'-54 $\Delta=2$

B:5 B':5

B+B'-58.5 $\Delta=6.5$

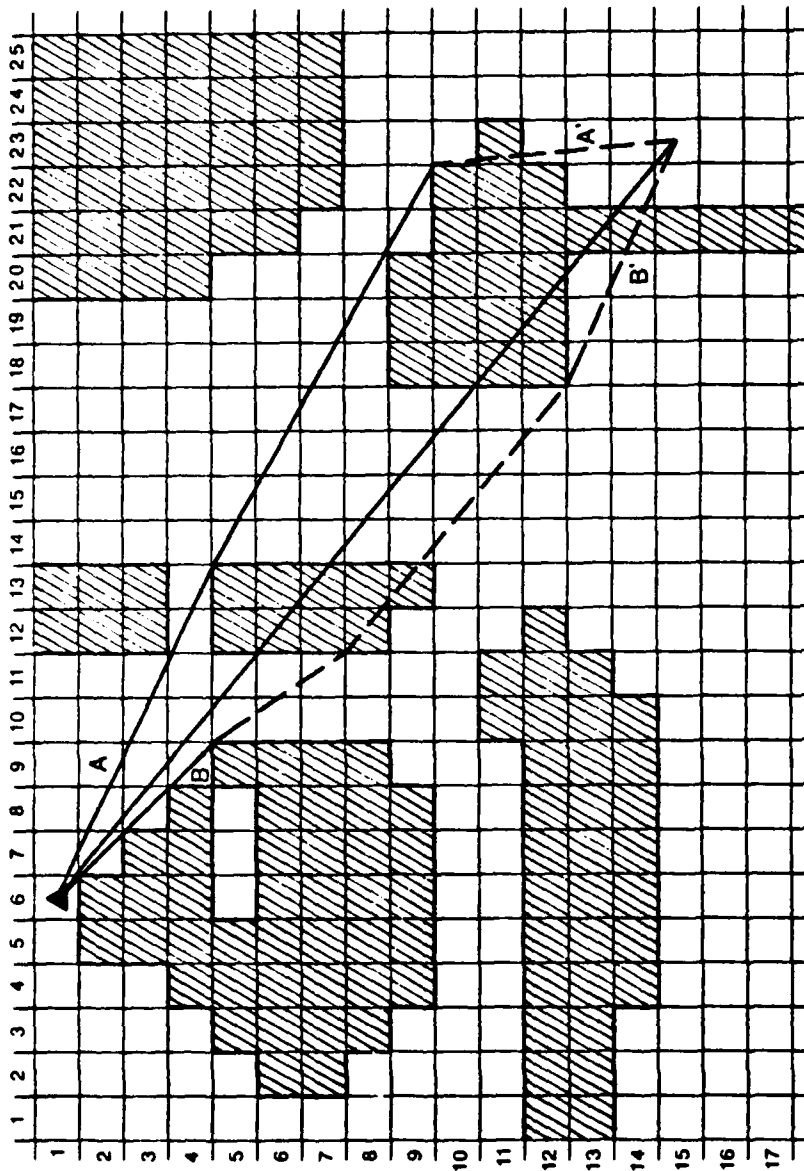
Figure 38. BINARY TRAVERSABILITY FOR FIRST GOAL

A last situation which did not imply extra but rather a ranking of classes was that of functor weight distribution. Weights were ranked between 0 and 10. Initially all classes were normalized to that range without regard for interclass relevance. That is, some decisions taken by the NAVIGATOR resulted questionable due to the wide overruling effect single classes, important to the situation or not, had in the whole process. It was obvious that some classes should influence decisions more than others, in lieu of arbitrary decision-making.

For example, to a given vehicle, terrain traversability may be of outmost priority than any distance or deviation to the goal. Beyond this, distance may be at a higher priority than deviation. Thus, the final class ranking for the vehicle simulation resulted in assigning a maximum of 10 to traversability in either passageway A or A', and a maximum of 8 to the A-A's composite distance range. Between A and A' deviations, it was, again, heuristically assumed that the latter's was more important than the former's. Thus a maximum of 6 was assigned to A', and of 4 to A.

Two trials were made for the NAVIGATOR. Two separate vocabularies were embedded in the Conceptron plane. The difference being that one was fuzzier (not as precise) than the other: 15 vs 30 input and 6 vs 15 output words, respectively. The classes remained at 5 input and 2 output in both trials. In addition, the more precise NAVIGATOR was given unlimited vision (full range of map where unobstructed) but was not run under an optimized Conceptron version. Neither of the differences made an exception, and as expected, the less fuzzier of the vocabularies just made more selective decisions along the way.

At the end, both routes coincided except at a very curious location in the map: the better informed vehicle decided to cross the shallow portion of the river rather than the dry bridge. This was attributed to the fact that the better informed NAVIGATOR (i.e., unlimited range of vision) could judge, a priori and from a far distance, on crossing the river. This would eventually result in a more efficient route (straight, point-to-point navigation). The poorly informed NAVIGATOR (i.e., limited range of vision) had to zig-zag over the bridge since its sensors could not capitalize on the potential point-to-point navigation opportunity supported by the existence of a shallow area in the river (Figures 39:42).



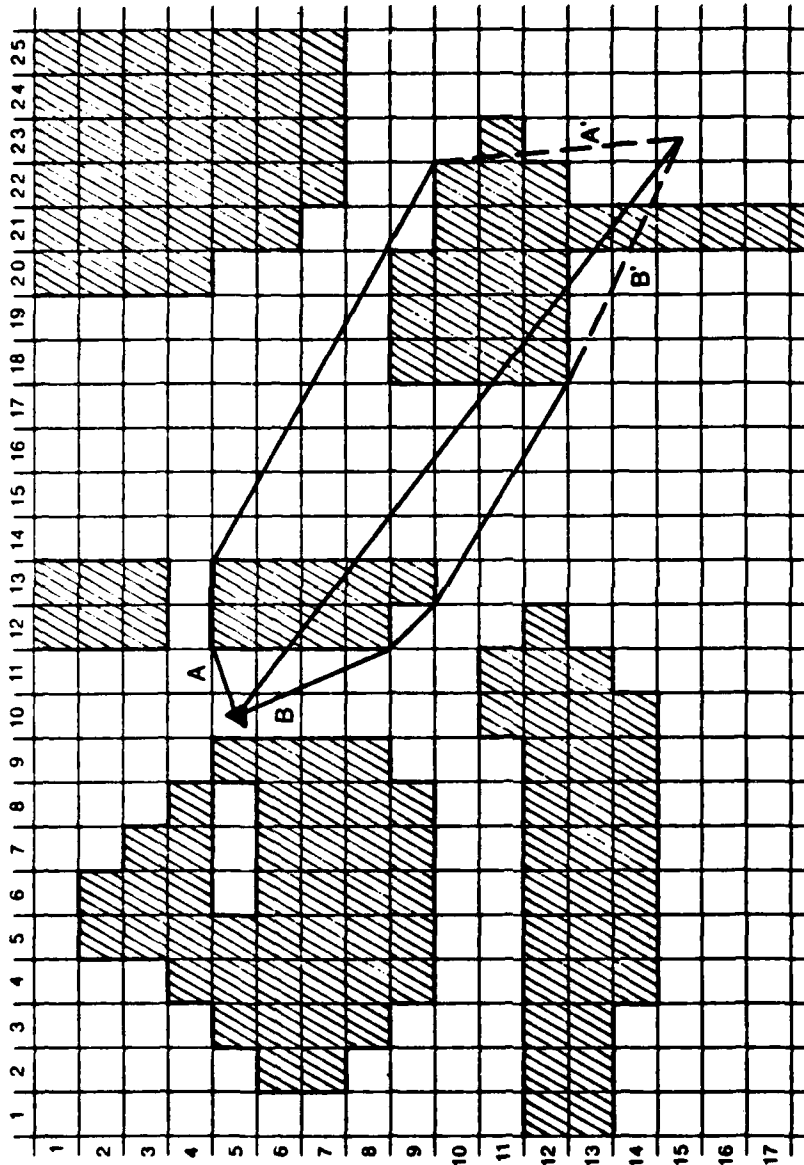
AVG. TRAVERSABILITY

RANGE-44

A: 5.27 A': 5
B: 5.00 B': 5

A+A': 48.5 Δ -4.5
B+B': 45 Δ -1

Figure 39. SECOND GOAL



AVERAGE DEVIATION		AVG. TRAV:		RANGE:33	
A: 18°	A': 48°	A: 5.18	A': 5	A+A': 38.5	Δ: 5.5
B, B' < 45°		B: 4.78	B': 5	B+B': 34.5	Δ: 1.5

Figure 40. THIRD GOAL

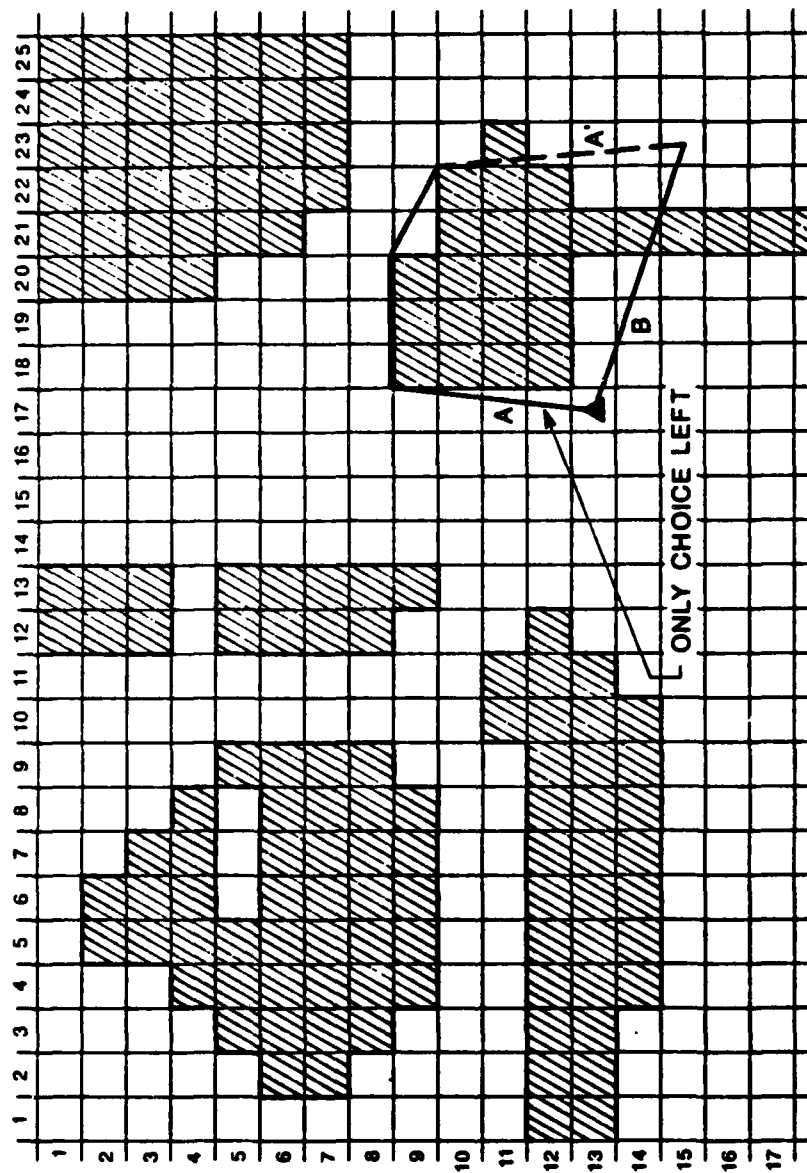
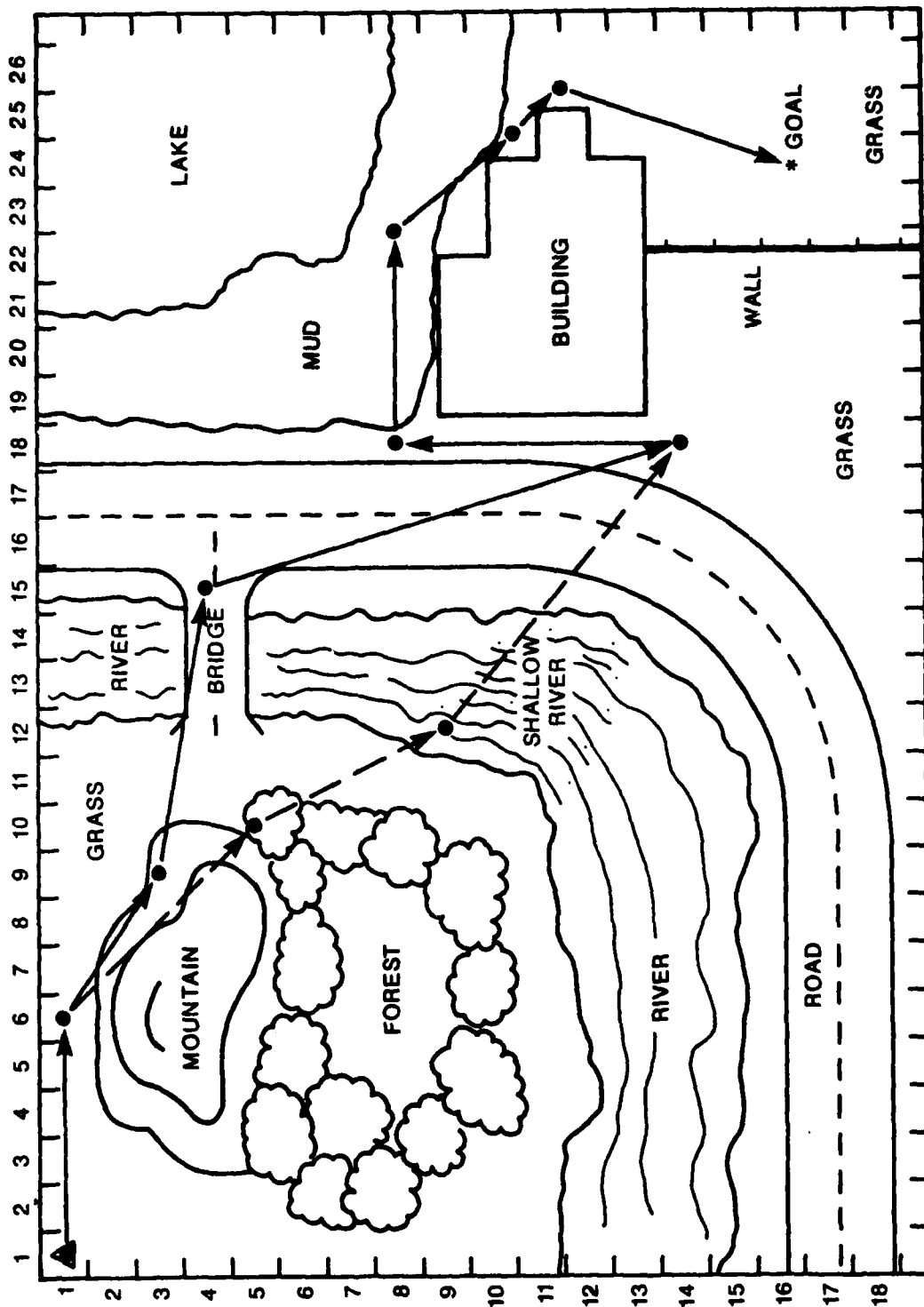


Figure 41. FINAL GOAL



6.0 CONCLUSION

Feasibility of a production-quality version of this vehicle theoretical prototype cannot be determined without the availability of suitable DBMT models like those presented in this paper (14). The NAVIGATOR design proposed here is preliminary and exploratory in form. However, its successful integration with a PILOT simulation was attempted and demonstrated. A real vehicle prototype was to be tested in late 1985/early 1986. Thus, reports on performance results should be expected in the near future.

The goal of an autonomous expert system was achieved with the prototype vehicle since unmanned perception was provided and the simulated mission was independent of any sort of remote control. Research is underway for the design of adaptive Conceptron/functor planes, including the evolution of the associative weights. This is an essential function required for mission effectiveness in dynamic environments where the vehicle must perform in stochastic mission envelopes. At least, undesirable vehicle performance degradation is eliminated via adaptive domains of expertise.

It is foreseen that the adaptation and learning will be carried out by each independent domain of expertise; that is, these functions will not be centralized. Inter-domain learning integrity will be supervised by the MANAGER. Accountability for temporal and spatial dynamics such as motion and role behavior will be performed by the tracking and forecasting constructs embedded in the domains. Heuristics will play a role in these constructs. Also, feedback mechanisms will be able to fine tune the Conceptron associations along satisfactory and consistent experience.

REFERENCES

1. N. Nilsson, Principles of Artificial Intelligence, Tioga, 1980.
2. A. Meystel and M. Thomas, "Computer-Aided Conceptual Design in Robotics," Proceedings of the IEEE International Conference in Robotics, Atlanta, Georgia, 1984.
3. A. Meystel, "Conceptron," Applied Systems and Cybernetics, Vol. II, G.E. Lasker, editor, from the Proceedings of the International Congress on Applied Systems Research and Cybernetics (Acapulco 1980), Pergamon Press, 1981.
4. A. Meystel, "Intelligent Control of a Multiactuator System," IFAC Information Control Problems In Manufacturing Technology, 1982, D.E. Hardt, editor, Pergamon Press, 1983.
5. A. Meystel, "Planning in the Anthropomorphic Machine Intelligence," Proceedings of the IEEE Conference on Cybernetics and Society, Seattle, WA, 1982.
6. P. Chen, "The Entity-Relationship Model -- Toward a Unified View of Data," ACM Transactions on Database Systems, Vol. I, No 1, 1976.
7. P. Chen, "English Sentence Structure and Entity-Relationship Diagrams," Information Sciences, Vol. 29, 1983.
8. C.J. Date, An Introduction to Database Systems, 3rd edition, Addison-Wesley, 1982.
9. R. Brooks, "Solving the Find-Path Problem by Good Representation of Free Space," IEEE Transaction on Systems, Man, and Cybernetics, SMC-13, 1983.
10. M. Ben-Bassat and A. Freedy, "Knowledge Requirements and Management in Expert Decision Support Systems for Military Situation Assessment," IEEE Transaction on Systems, Man, and Cybernetics, Vol. SMC-12, No 4, July/August, 1982.
11. J.T. Tou and R.C. Gonzalez, Pattern Recognition Principles, Addison-Wesley, 1974.
12. H. Herrlich and G. Strecker, Category Theory: An Introduction, Allyn & Bacon, 1973.
13. L.A. Zadeh, The Concept of a Linguistic Variable and Its Application to Approximate Reasoning - Parts I and II, Report ARO 10117.7-EL, UCLA Berkeley, 1975.
14. E.A. Martinez-Lopez, On Knowledge Bases for Production Systems, unpublished report, Computer Science Department, University of Florida, Gainesville, 1984.

BIBLIOGRAPHY

1. Brown, J.G. Fuzzy Sets on Boolean Lattices. US Army-Ballistic Research Laboratories. Memorandum Report 1957. January 1969.
2. Chen, P. "The Entity-Relationship Model -- Toward a Unified View of Data. ACM Transactions on Database Systems, Vol 1, No 1, 1976.
3. Date, C.J. An Introduction to Database Systems. 3rd edition. Addison-Wesley, 1982.
4. Dubois, D. and H. Prade. Fuzzy Sets and Systems: Theory and Applications. Academic Press, 1980.
5. Fu, K. and J. Tou, editors. Learning Systems and Intelligent Robots. Plenum Press, 1974.
6. Gardarin, G. and E. Gelenbe, editors. New Applications of Data Bases. Academic Press, 1984.
7. Goos, G. and J. Hartmanis, editors. Proceedings of the First International Symposium on Category Theory Applied to Computation and Control, February 1974. Springer-Verlag, 1975.
8. Herrlich, H. and G. Strecker. Category Theory: An Introduction. Allyn & Bacon, 1973.
9. Hurley, R.B. Decision Tables in Software Engineering. Van Nostrand Reinhold, 1983.
10. Isaacs, R. Differential Games. Wiley, 1967.
11. Kandel, A. and S. Lee. Fuzzy Switching and Automata: Theory and Applications. Crane & Russak, 1979.
12. Kaufmann, A. Theory of Fuzzy Subsets. Volume I. Academic Press, 1975.
13. Kent, W. Data and Reality. North-Holland, 1978.
14. Korfhage, R.R. Discrete Computational Structures. Academic Press, 1984.
15. Mamdani, E.H. and B.R. Gaines, editors. Fuzzy Reasoning and Its Applications. Academic Press, 1981.
16. Meystel, A. and M. Thomas. "Computer-Aided Conceptual Design in Robotics." Proceedings of the IEEE International Conference in Robotics. Atlanta, Georgia. 1984.
17. Meystel, A. "Conceptron." Applied Systems and Cybernetics. Volume II. G.E. Lasker, editor. From the Proceedings of the International Congress on Applied Systems Research and Cybernetics (Acapulco 1980). Pergamon Press, 1981.

BIBLIOGRAPHY (CONTINUED)

18. Meystel, A. "Intelligent Control of a Multiactuator System." IFAC Information Control Problems In Manufacturing Technology, 1982. D.E. Hardt, editor. Pergamon Press, 1983.
19. Negoita, C. Applications of Fuzzy Sets to Systems Analysis. Wiley.
20. Nilsson, N. Principles of Artificial Intelligence. Tioga, 1980.
21. Pattee, H., editor. Hierarchy Theory: The Challenge of Complex Systems. Braziller, 1973.
22. Pierce, R.S. Associative Algebras. Springer-Verlag, 1982.
23. Rubinstein, M.F. Patterns of Problem-Solving. Prentice-Hall, 1975.
24. Sime, M.E. and M.J. Coombs, editors. Designing for Human-Computer Communication. Academic Press, 1983.
25. Spriet, J.A. and G.C. Vansteenkiste. Computer-Aided Modelling and Simulation. Academic Press, 1982.
26. Tou, J.T. and R.C. Gonzalez. Pattern Recognition Principles. Addison-Wesley, 1974.
27. Uhr, L. Algorithm-Structured Computer Arrays and Networks. Academic Press, 1984.
28. Waterman, D.A. and F. Hayes-Roth. Pattern-Directed Inference Systems. Academic Press, 1978.
29. Wee, W.G. On Generalization of Adaptive Algorithms and Application of the Fuzzy Sets Concept to Pattern Classification. Doctoral Dissertation in Electrical Engineering. Purdue University, 13 July 1967.
30. Zadeh, L.A. The Concept of a Linguistic Variable and Its Application to Approximate Reasoning - Parts I and II. Report ARO 10117.7-EL. UCLA-Berkeley, 1975.
31. Zadeh, L.A. A Fuzzy-Algorithmic Approach to the Definition of Complex or Imprecise Concepts. Memo ERL-M474. UCLA-Berkeley, 11 October 1974.
32. Zadeh, L.A. Fuzzy Logic and Approximate Reasoning. Memo ERL-M479. UCLA-Berkeley, 12 November 1974.
33. Zadeh, L.A. Fuzzy Sets. Report 64-44. UCLA-Berkeley, 16 November 1964.
34. Zadeh, L.A., et al, editors. Fuzzy Sets and Their Applications to Cognitive and Decision Processes. Academic Press, 1974.
35. Zadeh, L.A. "Outline of a New Approach to the Analysis of Complex Systems and Decision Processes". IEEE Transcription on Systems, Man, and Cybernetics. Volume SMC-3, No. 1. January 1973.